

Raport projektowy z przedmiotu CPSB

URZĄDZENIE DO BEZPRZEWODOWEGO POMIARU
EKG

JAKUB SZYMAŃSKI HUBERT KORZENIEWSKI

https://github.com/hkorzeniewski/ECG-ESP32-tool/tree/main/full_app

1. Założenia projektu

Założeniem projektu jest stworzenie bezprzewodowego urządzenia umożliwiającego pomiar sygnału EKG człowieka. W tym celu użyjemy płytki ESP32_DevKitC z wbudowanym modułem ESP32-SOLO-1 i czujnika monitorującego sygnał serca SEN0213. Moduł ESP32 będzie miał za zadanie komunikować się za pomocą sieci WiFi z aplikacją (komputerem), która będzie odbierać sygnał i poddawać go odpowiedniej analizie. W wyniku działania programu otrzymamy odpowiednie informacje na temat pomiaru.

2. Część sprzętowa

W związku z tym, że główny czujnik układu został pożyczony na potrzeby wykonania projektu, urządzenie zmontowane jest na „pajką” za pomocą krótkich przewodów. Cała część sprzętowa składa się z trzech elementów:

1. Płytką ESP32_DevKitC z modułem ESP32-SOLO-1



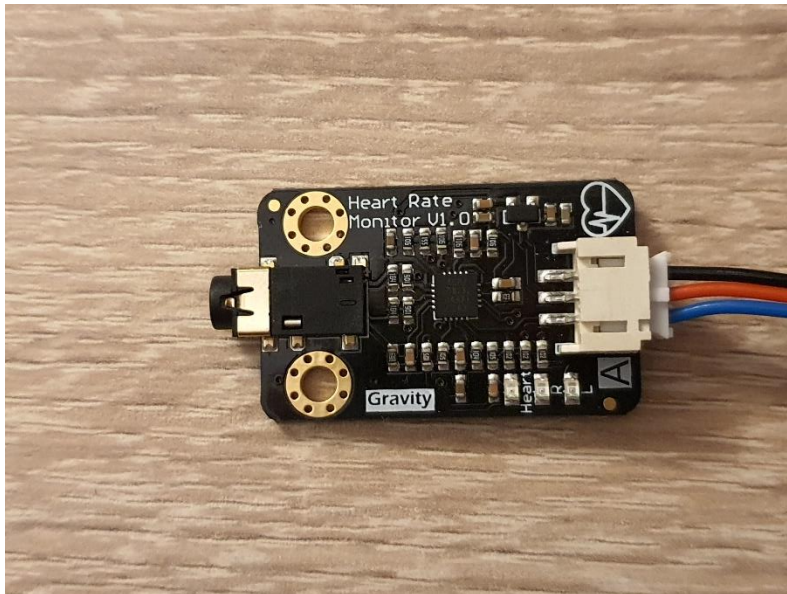
ESP32 jest układem typu SoC produkowanym przez chińską firmę *Espressif Systems*. Układ ten pozwala na komunikację przez WiFi oraz przez Bluetooth. Powyższe zdjęcie przedstawia układ ESP32 w module DevKitC, który posiada 38 pinów odpowiadających wyprowadzeniom układu.

Dokładna dokumentacja modułu ESP32-SOLO-1:

https://www.espressif.com/sites/default/files/documentation/esp32-solo-1_datasheet_en.pdf

W naszym projekcie układ ESP32 pełni rolę serwera. Kod serwera został napisany w środowisku Arduino IDE. Żeby mieć możliwość programowania ESP32 z platformy Arduino wcześniej musieliśmy zainstalować odpowiednie biblioteki oraz podać dokładną specyfikację płytki. W kodzie serwera ustawiamy najważniejsze parametry służące do odczytu informacji z analogowego wejścia i komunikacji między serwerem a klientem.

2. Czujnik monitorujący sygnały z serca SEN0213



Jest to układ pozwalający w dość prosty sposób odebrać sygnały z serca, które odczytujemy analogowo z trzech elektrod przyczepionych do ciała pacjenta. Elektrody podłączone są do układu za pomocą gniazda mini jack (widoczne z lewej strony układu). Zasilanie układu równe 5V pobierane jest z układu ESP32. Wyjście „data” także jest podłączone do modułu ESP32 do jednego z pinów odczytujących wyjście analogowe.

Pełna specyfikacja SEN0213:

- Input Voltage: 3.3 - 6V (5V recommended)
- Output Voltage: 0 - 3.3V
- Operating current: <10mA
- Dimension: 35 x 22(mm), 1.378" x 0.866"(in)
- Interface Type: PH2.0-3P

3. Źródło zasilania – powerbank

W naszym projekcie ze względów bezpieczeństwa pomiary nie mogą być wykonywane przy sprzęcie podpiętym bezpośrednio do sieci elektrycznej. W związku z tym jako zasilanie części sprzętowej użyliśmy powerbanka, który na wyjściu ma odpowiednio duże napięcie pozwalające na zapewnienie stosunkowo długiej i stabilnej pracy układu.

3. Opis działania sprzętu

Układ modułu ESP32 wraz z czujnikiem SEN0213 komunikuje się z komputerem przy użyciu sieci WiFi. Serwerem jest moduł ESP32, do którego możemy podłączyć klienta za pomocą odpowiedniej aplikacji z komputera.

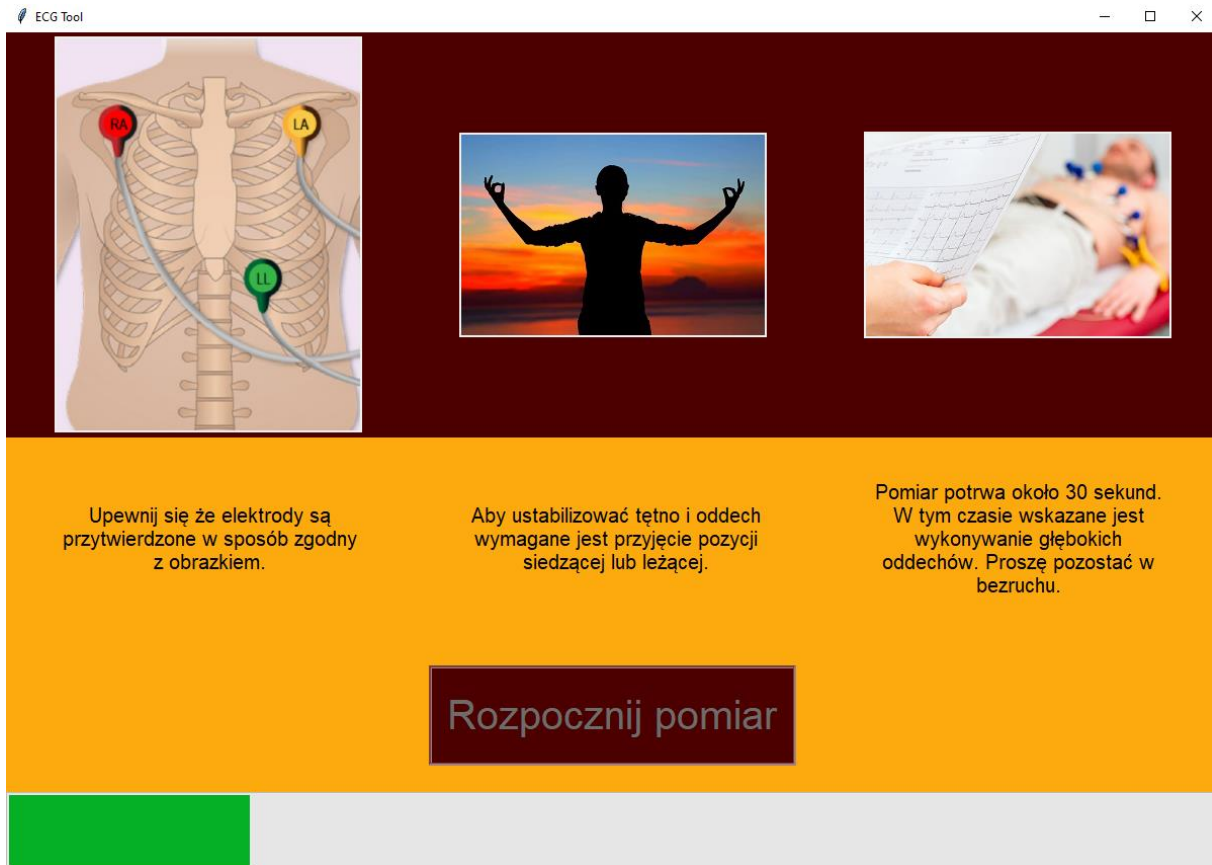
Kod serwera został napisany w środowisku Arduino, języku opartym na C. W kodzie ustawiamy parametry sieci z której chcemy korzystać, port po którym będziemy się komunikować, analogowy pin układu, z którego będziemy odczytywać dane oraz liczbę próbek, którą pobierzemy w pojedynczym pomiarze. Na potrzeby wykonania projektu wszystkie te dane są zapisane „sztywno” w kodzie. Po włączeniu zasilania serwer „oczekuje” na zgłoszenia od klientów na odpowiednim porcie i IP.

Kod klienta został napisany w języku Python przy użyciu socketów. Ustawiając odpowiednie IP i port, otwieramy połączenie między klientem i serwerem. Po uzyskaniu połączenia klient zaczyna „pobierać” dane z serwera i zaczyna się pomiar. W trakcie pomiaru dane zapisywane są do listy, która przedstawia odczyt w określonym czasie. Dane te będą wykorzystywane w dalszej części projektu, która jest odpowiedzialna za analizę.

4. Opis oprogramowania aplikacji desktopowej



Po uruchomieniu programu, oczom użytkownika ukazują się obraz służący do wstępnego zapoznania się z aplikacją. Kolejne okno zawiera instrukcję użytkową programu. Opisuje jakie czynności wykonać, by pomyślnie przeprowadzić pomiar. Ostatni obraz przedstawia wyniki pomiaru w postaci tętna(bpm) i współczynnika oddechu(częstotliwość oddechu w czasie 60 sekund). Wygenerowany zostaje także wykres obrazujący część próbek sygnału.



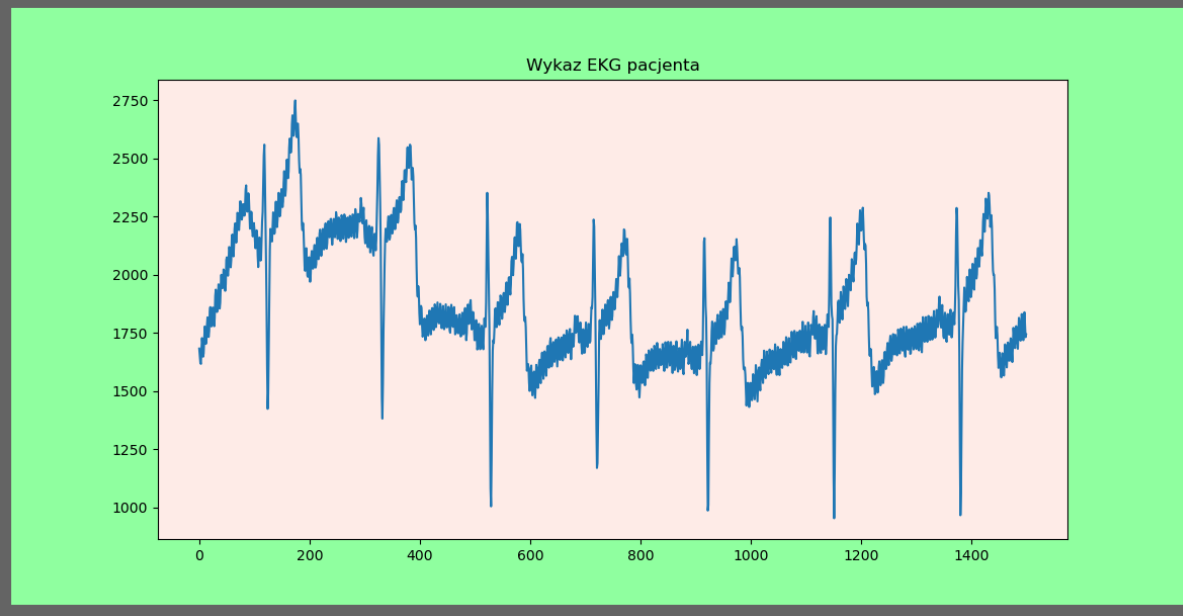
5. Implementacja aplikacji desktopowej

Kod aplikacji desktopowej został napisany w języku Python. W celu wygenerowania przejrzystego interfejsu użytkownika GUI(ang. Graphical User Interface), zastosowano bibliotekę Tkinter. Architektura programu opiera się na prostym i jakże skutecznym wzorcu projektowym MVC(model-view-controller). Zapewnia on doskonały podział funkcjonalności w programie. Klasa Model odpowiada za obsługę danych. W tym przypadku uruchamia kod aplikacji klienckiej służącej do pobierania próbek sygnału EKG z serwera. Zajmują się także filtracją i analizą otrzymanych danych. Klasa View odpowiada za interakcję z użytkownikiem aplikacji, oraz implementuje jej finalny wygląd. Klasa Controller zwiera logikę wyżej wymienionych klas, zapewniając poprawne działanie aplikacji.

Wyniki badań

Tętno: 71, norma 60~100

Współczynnik oddechu: 13, norma 12~16



6. Cyfrowe przetwarzanie sygnału biologicznego

Sygnał EKG próbkowany jest z częstotliwością 250 Hz, co daje 4 ms okresu próbkowania. Otrzymany pomiar świetnie obrazuje elektrokardiogram. W celu analizy tętna i innych współczynników medycznych, zastosowany zostaje filtr środkowo-zaporowy wycinający szumy o częstotliwości wynoszącej 0.05 Hz. Do wykrycia próbek w punkcie R sygnału EKG, wykorzystano bibliotekę `heartpy` (<https://python-heart-rate-analysis-toolkit.readthedocs.io/en/latest/>). Aby zidentyfikować bicie serca, obliczana jest średnia ruchoma w oknach czasowych wynoszących 0.75 s. Maksimum obszaru którego amplituda jest większa, niż wcześniej wyliczona średnia jest znanym nam punktem R sygnału ECG. Jedną z interesujących właściwości serca jest to, że na częstotliwość, z jaką bije silnie wpływa oddychanie, poprzez autonomiczny układ nerwowy. Jest to jeden z powodów, dla których głębokie oddychanie może uspokoić nerwy. Zależność ta została także zastosowana do wyodrębnienia częstotliwości oddechów.

7. Bibliografia i źródła wiedzy

<https://www.dfrobot.com/product-1510.html>

<https://forbot.pl/blog/leksykon/esp32>

<https://python-heart-rate-analysis-toolkit.readthedocs.io/en/latest/>

<https://docs.python.org/3/library/tkinter.html>