

# Projekt EIUA - wykrywanie fazy REM za pomocą czujnika tętna ludzkiego

Jakub Pawlak  
Jakub Słabicki  
Michał Wróblewski

17 stycznia 2018

# Spis treści

<b>1</b>	<b>Cel projektu</b>	<b>3</b>
<b>2</b>	<b>Główne założenia i wstęp teoretyczny</b>	<b>3</b>
2.1	Wstęp teoretyczny . . . . .	3
2.1.1	Faza REM . . . . .	3
2.1.2	Użyte komponenty . . . . .	3
2.1.3	Aplikacja na telefon . . . . .	3
2.2	Główne założenia . . . . .	4
<b>3</b>	<b>Opis działania modułu i aplikacji</b>	<b>4</b>
3.1	Opis techniczny i zasada działania modułu do pomiaru tętna . . . .	4
3.2	Opis działania komunikacji . . . . .	5
3.2.1	Opis ogólny . . . . .	5
3.2.2	Opis protokołu . . . . .	6
3.3	Opis działania aplikacji . . . . .	7
<b>4</b>	<b>Schematy ideowe</b>	<b>11</b>
<b>5</b>	<b>Testy, wyniki prac</b>	<b>15</b>
<b>6</b>	<b>Podsumowanie</b>	<b>18</b>
<b>7</b>	<b>Bibliografia</b>	<b>19</b>

# 1 Cel projektu

Celem projektu było stworzenie od podstaw pulsometru - urządzenia do monitorowania pracy serca. W projekcie tym należało skorzystać z opcji z opaską na nadgarstek. W opasce znalazły się wszelkie czujniki i komponenty do wysyłania przez bluetooth informacji o pulsie. Drugim zadaniem było napisanie aplikacji (na Androida), w której to odbierany miał być sygnał z opaski. Dalszą częścią było stworzenie algorytmu do wykrywania fazy REM oraz implementacja innych ważnych funkcji do dobrego działania aplikacji.

## 2 Główne założenia i wstęp teoretyczny

### 2.1 Wstęp teoretyczny

#### 2.1.1 Faza REM

Faza REM to taka faza snu, w której wyczuwalny jest znaczny wzrost tętna ludzkiego oraz ruchu gałek ocznych. I od tego też wzięła się nazwa tej fazy - z ang. Rapid Eye Movement. REM występuje w czasie snu mniej więcej co 90 minut, dlatego wiele ludzi ustawia sobie czas spania jako wielokrotność tych 90 min. Pobudka w czasie tej fazy jest bardzo korzystna dla człowieka. Przede wszystkim poprzez wcześniejszy wzrost tętna, po obudzeniu się ludzie są bardziej zmobilizowani do życia, aktywniejsi. Przejście między spaniem a pobudką jest łagodniejsze. Co więcej, udowodnione zostało, że to właśnie w tej fazie snu śnimy najwięcej rzeczy, więc obudzenie się w tym oknie czasowym sprawia, że człowiek jest w stanie pamiętać swoje sny.

#### 2.1.2 Użyte komponenty

- Moduł optoelektronicznego czujnika pulsu PS01
- Moduł do wysyłania Bluetooth DFR0339
- Zasilanie

#### 2.1.3 Aplikacja na telefon

Drugim zadaniem projektu było stworzenie aplikacji na telefon. Wybrane zostały telefony z oprogramowaniem Android. Sama zaś aplikacja została napisana w języku programistycznym Java, przy użyciu środowiska Android Studio. W aplikacji oprócz samego wyboru budzika do wykrywania fazy REM zostały zaimplementowane również inne funkcje, które zostaną opisane w późniejszych punktach

## 2.2 Główne założenia

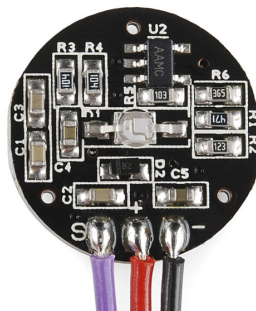
Projekt miał 3 główne założenia:

- Złożenie opaski złożonej z wyżej wymienionych komponentów. Czujnik pulsu miał wysyłać przez bluetooth sygnał dzięki któremu można było uzyskać tętno człowieka.
- Stworzenie aplikacji, w której to uzyskiwany został sygnał z bluetooth.
- Połączenie ze sobą aplikacji i modułu do odczytywania pulsu. Testowanie i nanoszenie poprawek.

## 3 Opis działania modułu i aplikacji

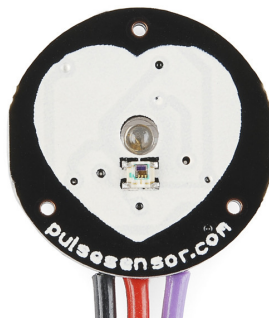
### 3.1 Opis techniczny i zasada działania modułu do pomiaru tętna

Do projektu użyliśmy moduł optoelektronicznego czujnika pulsu PS01. Działa on pod napięciem do 5V. W module zastosowano wzmacniacz i eliminator szumów, dzięki czemu moduł jest precyzyjny i niezawodny.



Rysunek 1: Widok czujnika z jeden strony

Po przyłożeniu czujnika do ciała na wyjściu pojawia się sygnał analogowy, który po przeliczeniu przez algorytm, który znajduje się w bibliotece od niego, otrzymujemy obecny poziom pulsu. To wszystko dzięki temu, że czujnik wykrywa charakterystyczne zmiany w tkankach poprzez analizę odbitego światła.

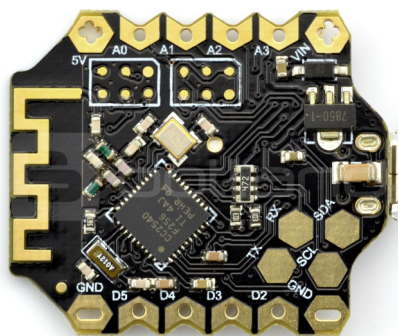


Rysunek 2: Widok czujnika z drugiej strony

## 3.2 Opis działania komunikacji

### 3.2.1 Opis ogólny

Rodzaj połączenia jakie potrzebowaliśmy to bezprzewodowe, zdecydowaliśmy się więc na użycie Bluetooth Low Energy lub inaczej zwany Bluetooth Smart. Czujnik pulsu jest połączony z mikrokontrolerem Bluno Beetle, który okazał się najtańszą i najprostszą opcją układu z obsługą BLE (Bluetooth Low Energy). Inne układy wymagały specjalnych programatorów co znacznie podwyższało poziom trudności programowania oraz koszty. Bluno Beetle oparty na ATmega328 oraz układzie CC2540 obsługującym Bluetooth LE, pozwolił nam na obsługę czujnika pulsu i przesłanie wartości obliczonego pulsu do smartphon'a. Całość również może być zasilana napięciem mniejszym niż 5V, przez co idealnie pasuje tutaj użycie akumulatora Li-Pol z jednym ogniwem, ponadto rozmiary płytki PCB to 33 x 29 mm co jest dobre do naszego zastosowania, gdzie liczy się objętość całej opaski.



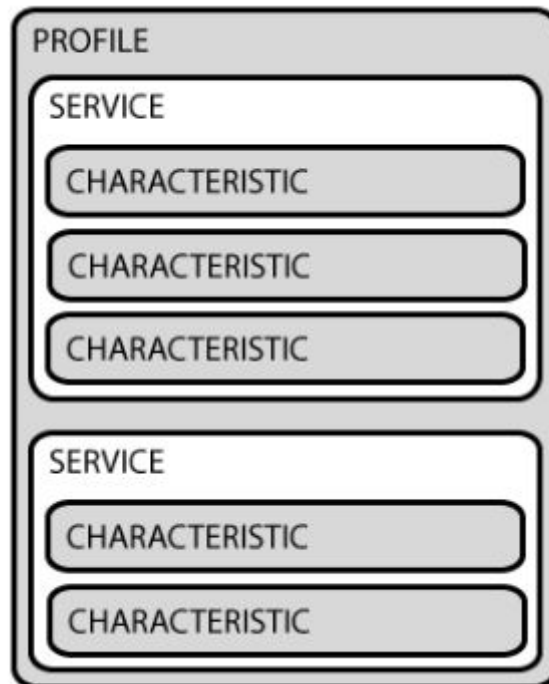
Rysunek 3: poglądowy wygląd układu

Kolejnym argumentem, który stoi za wyborem tego układu jest port micro-USB, który jest bardzo popularny, przez co umożliwia programowanie w domowych

warunkach oraz to, że widoczny jest w systemie jako Arduino Uno - pozwala to nam na wykorzystanie dużej ilości bibliotek oraz oprogramowania, m.in. podglądu wejść/wyjść co ułatwiło testowanie czujnika pulsu PS01. W punkcie poświęconym schematom ideowym zamieszczony jest schemat elektryczny tegoż układu.

### 3.2.2 Opis protokołu

Bluetooth BLE działa w pasmie częstotliwości 2.400 - 2.4835 GHz. Jego zasięg to ponad 100m, a średni pobór prądu urządzeń wykorzystujących ten protokół to  $< 15\text{mA}$ , gdzie przy klasycznym Bluetooth'e było to około 30mA. Podstawą tego protokołu są transakcje GATT oparte o serwer GATT. Zawiera on obiekty tzw. profile, te zawierają kolejne obiekty nazywane serwisami, które posiadają 16 bitowe tablice, każda z przyporządkowanym ID, tablice te nazywane są charakterystykami. Schematyczne działanie protokołu przedstawia Rysunek 4



Rysunek 4: schematyczne przedstawienie protokołu

- profil jest obiektem abstrakcyjnym, tak naprawdę nie jest zdefiniowany, jest to po prostu zbiór serwisów. np Profil 'Heart Rate' zawiera 'Heart Rate Service' oraz 'Device Information Service'.

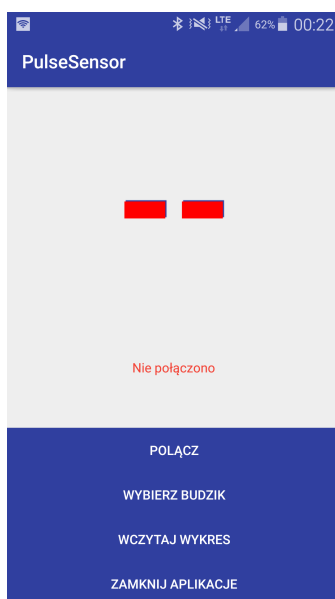
- serwis jest używana do upakowywania danych w logiczne całości, zawiera w sobie charakterystyki, każdy serwis ma unikalny 16bitowy ID zwany UUID
- charakterystyki - jest to najniższy poziom w definicji GATT, zawierający w sobie już rzeczywiste dane np. odczyty z czujników, każda charakterystyka ma również swój unikalny 16 bitowy ID - UUID
- deskryptory - ponadto każda charakterystyka może posiadać go, zawiera on informację na temat danych np. ich jednostkę, wtedy w charakterystyce odczytujemy wartość, a deskryptor pozwala nam określić jaka jest to jednostka

Aby komunikacja działała muszą zostać zaimplementowane następujące komendy, które klient GATT użyje aby odpytać serwer GATT: - Znajdź serwisy i ich UUID - Znajdź wszystkie charakterystyki dla danego serwisu - Znajdź charakterystykę o danym UUID - Odczytaj wartość z charakterystyki - Znajdź deskryptory do danej charakterystyki - Odczytaj deskryptory Te funkcjonalności zostały zaimplementowane w bibliotekach, z których skorzystaliśmy i dostosowaliśmy do własnych potrzeb. Ponadto musieliśmy uzyskać dostęp do adaptera Bluetooth z poziomu smartphon'a. To wszystko składa się na nasz protokół transmisji pomiędzy mikrokontrolerem Bluno Beetle a smartphon'em.

### 3.3 Opis działania aplikacji

Implementacja kodu została podzielona na kilka głównych etapów, a zadania zostały rozdzielone. Równocześnie każdy z projektantów mógł pracować nad aplikacją. Aplikacja działa w następujący sposób. Gdy otwieramy aplikację, gdzie należy wybrać przycisk Połącz, gdzie sprawdzane jest połączenie bluetooth. Jeżeli użytkownik połączy się z opaską, która ma na nadgarstku, w czasie rzeczywistym będzie mógł widzieć na jakim poziomie jest jego tętno. Użyte do tego zostało pole EditText w którym to stworzony został wątek z pętlą do odczytywania pulsu. Następnie udołu są 4 przyciski (Button), które zostaną szczegółowo omówione. Widok menu głównego przedstawia rysunek nr 5

- Połącz - naciskając ten przycisk, otwierana jest nowa aktywność, która ma za zadanie połączyć telefon z czujnikiem bluetooth. Z poziomu aplikacji użytkownik wybiera, z którym portem chce się połączyć. Następnie wyświetlana jest informacja o tętnie, oraz szczegółowe dane odnośnie uzyskanych danych w TextView. Dodana została opcja ScrollView. Tę aktywność można zobaczyć na rysunku nr 6.
- Wybierz Budzik - Aplikacja daje do wyboru użytkownikowi jaki chce budzik. Po naciśnięciu przycisku zostaje przekierowany do nowej aktywności,



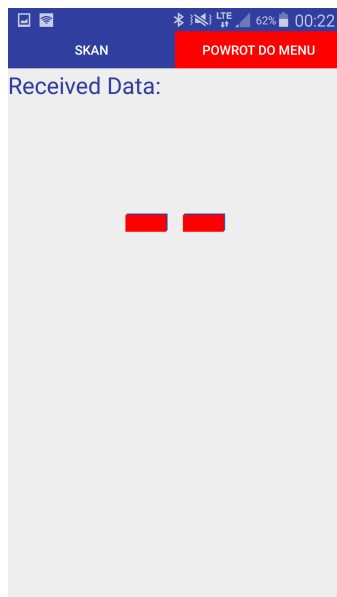
Rysunek 5: Widok na menu główne

w której są dwa przyciski - Zwykły budzik, Budzik REM. Z obydwu opcji jest otwierane nowe okno. Aktywność ta pokazana jest na rysunku nr 7

1. Zwykły budzik:

Pierwsza z opcji otwiera nam aktywność, gdzie użytkownik wybiera godzinę pobudki (okno TimePicker). Następnie za pomocą przycisku ustaw, tworzymy nowy intent oraz wyświetlana jest informacja na temat długości spania w TextView. Ponadto jest możliwość anulowania budzika przy naciśnięciu przycisku Anuluj. Alarm jest usuwany dzięki użyciu funkcji cancel z biblioteki AlarmManager. Skorzystano z klasy, która dziedziczy po klasie BroadcastReceiver. Gdy nadejdzie godzina pobudki, nawet przy zabitej aplikacji, włącza nową aktywność, w której jest możliwość zatrzymania budzika. Sam Alarm działa dzięki użyciu klasy MediaPlayer. Aby być pewnym, że na pewno będzie dźwięk, skorzystano zostało z AudioManager, w której regulujemy głośność. Została ona ustawiona na  $0.8 * \text{maksymalnej głośności}$ . Po ustaniu budzika głośność multimediiów zostaje przywrócona do stanu sprzed budzika. Automatycznie również otwiera się wykres w GraphView. Niestety na dzień 17.01 nie można zobaczyć przebiegu wykresy tętna. Następnie możemy przejść do menu głównego aplikacji wybierając przycisk "Powrót do menu".

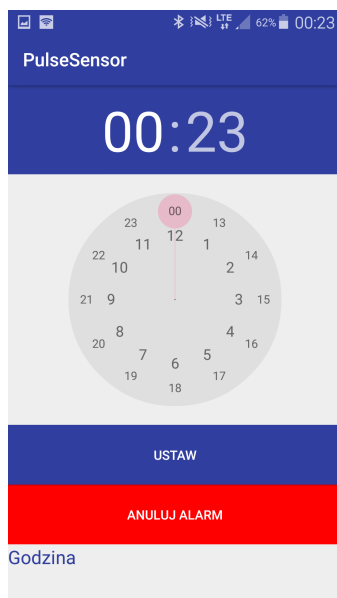




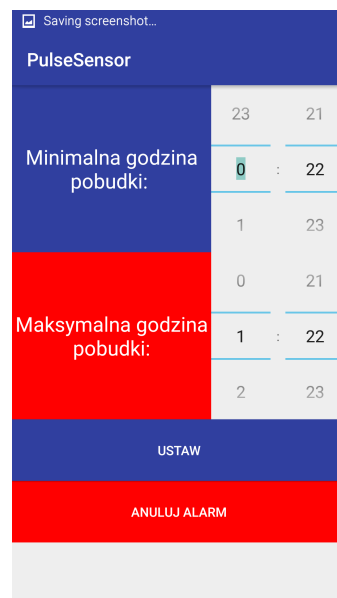
Rysunek 6: Aktywnosc 'połącz'



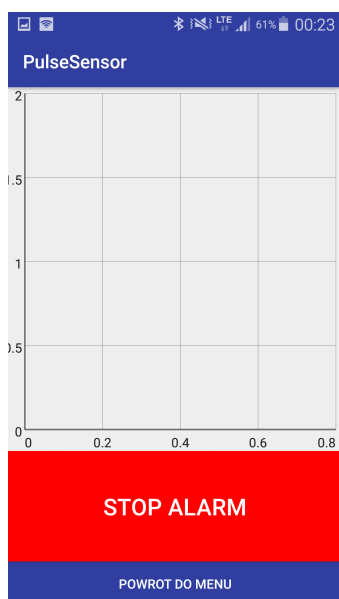
Rysunek 7: Wybór budzika



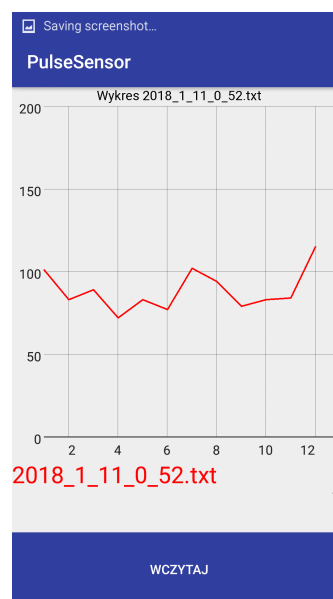
Rysunek 8: Zwykły Budzik



Rysunek 9: Budzik REM



Rysunek 10: Widok włączonego



Rysunek 11: Wczytaj wykres

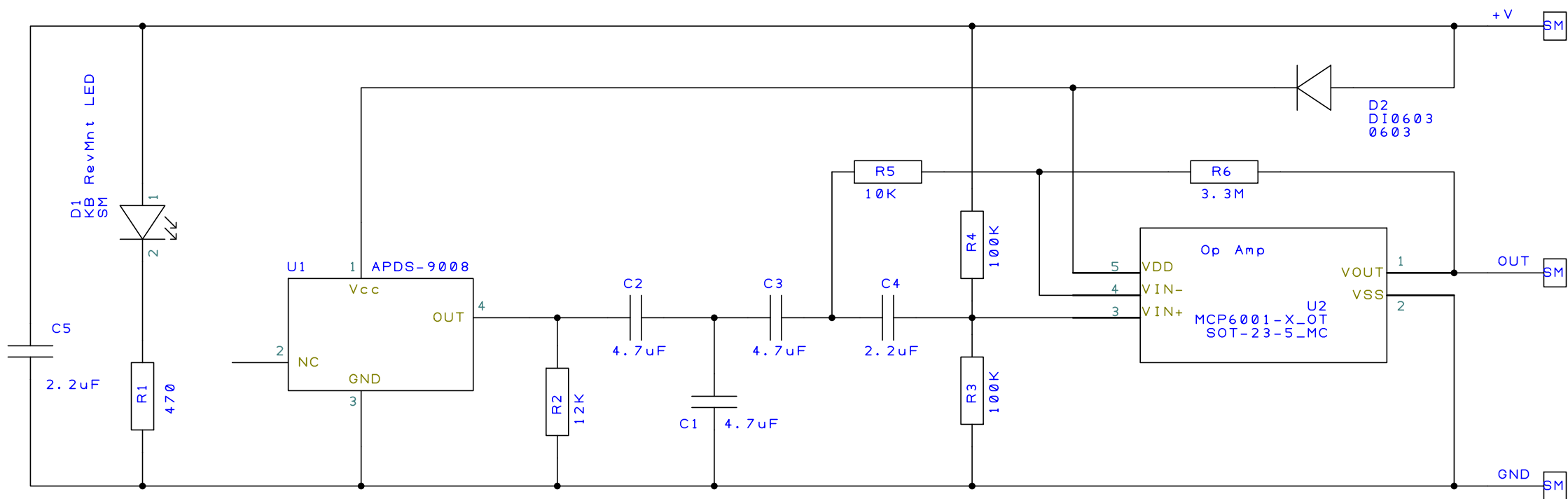
## 2. Budzik REM:

Wybór "Budzik REM" przekierowuje nas do nowej aktywności. Użytkownik ma do wyboru okno czasowe, którym chce się obudzić. Działa to dzięki użyciu NumberPicker. Użytkownik ma do wyboru godzinę i minutę minimalnego czasu spania oraz maksymalnego czasu spania. Gdy wybije godzina pierwsza, włączana zostaje klasa, która tak jak wcześniej dziedziczy po BroadcastReceiver. Włączony zostaje wątek, w którym jest autorski algorytm do wykrywania fazy REM. Jeżeli zostaje wykryta, włączony jest nowy intent, który włącza alarm (Rysunek nr 10). Aktywność ta jest identyczna jak przy zwykłym budziku. Jeżeli nie zostaje wykryta faza REM, alarm zostaje włączony równo z maksymalną godziną pobudki.

- Wczytaj Wykres - naciskając ten przycisk użytkownik przechodzi do nowej aktywności. Dzięki użyciu Spinner użytkownik może wybrać i zobaczyć na wykresie wcześniej zapisane dane. Dane te są przechowywane w folderze PulseSensor w pamięci wewnętrznej telefonu, który został utworzony przy pierwszym włączeniu aplikacji (sprawdzone jest każdorazowo przy włączeniu aplikacji czy jest utworzony dany folder - jeżeli nie, to z poziomu aplikacji jest tworzony). Po wybraniu przez użytkownika daty, należy nacisnąć przycisk Wczytaj Wykres i dzięki GraphView zostaje narysowany wykres z danych z pliku tekstowego. Aktywność tę zobaczyć można na rysunku nr 11

- Zamknij Aplikacje - Przycisk zamyka aplikacje. Skorzystane tutaj zostało z funkcji `finish()`, oraz `System.exit(0)`;

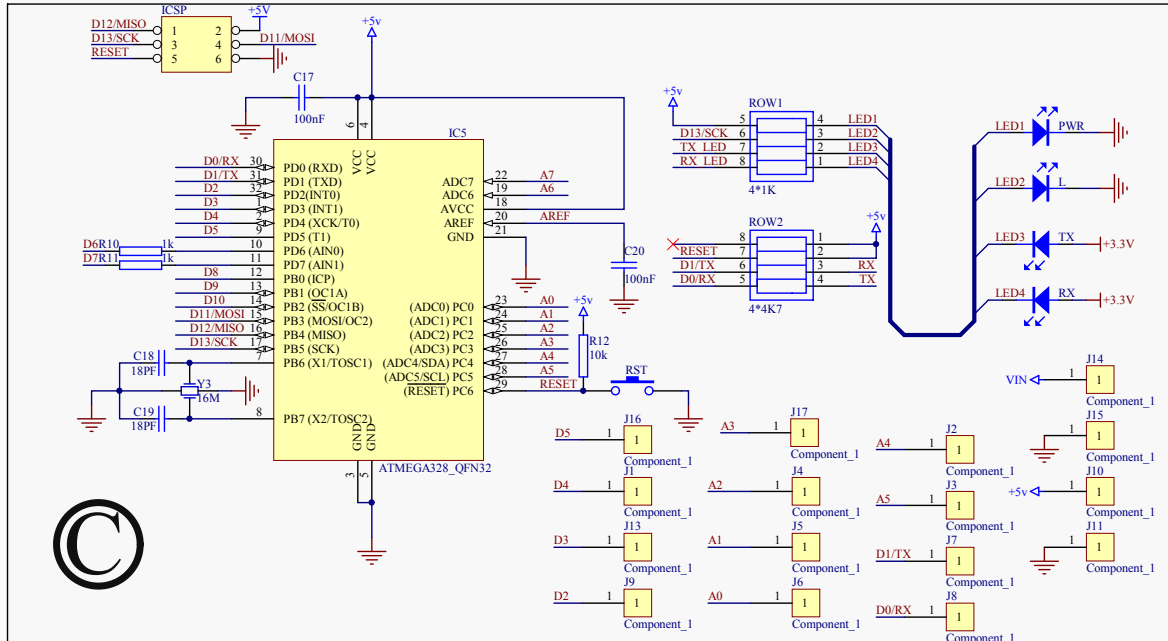
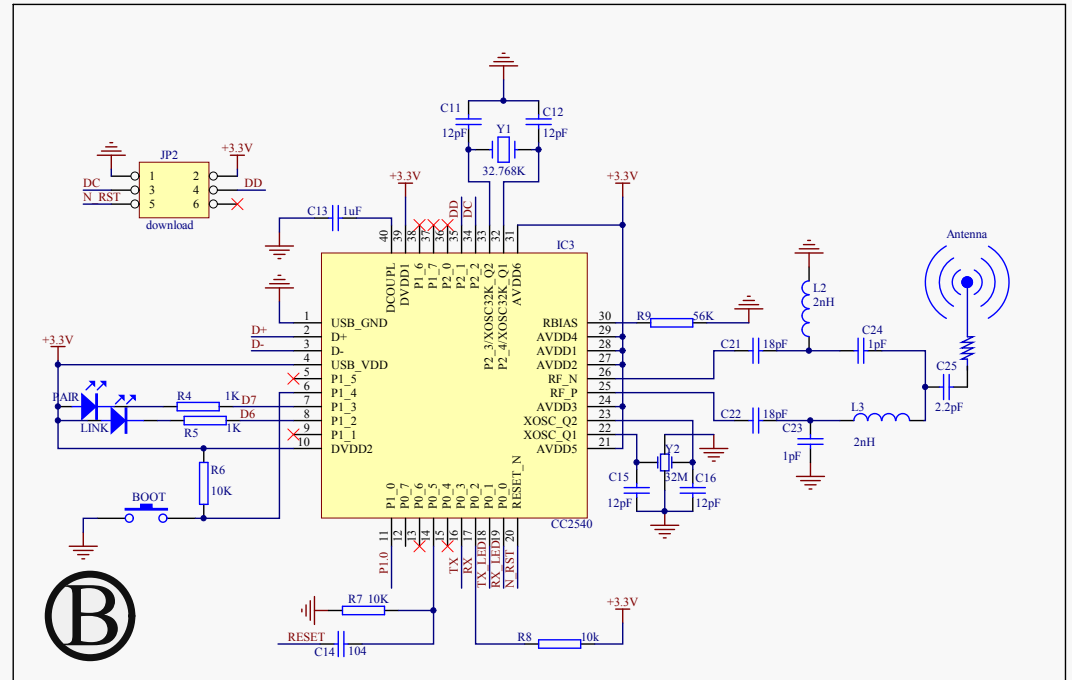
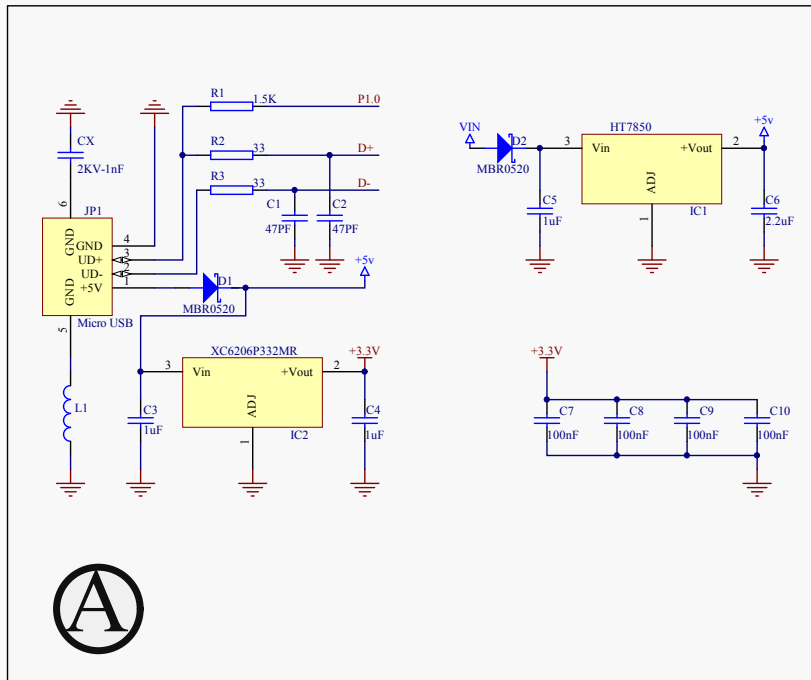
## 4 Schematy ideowe



Pulse Sensor Amplified

Designed by Joel Murphy  
Spring 2012

Licensed under the TAPR Open Hardware License ([www.tapr.org/OHL](http://www.tapr.org/OHL))

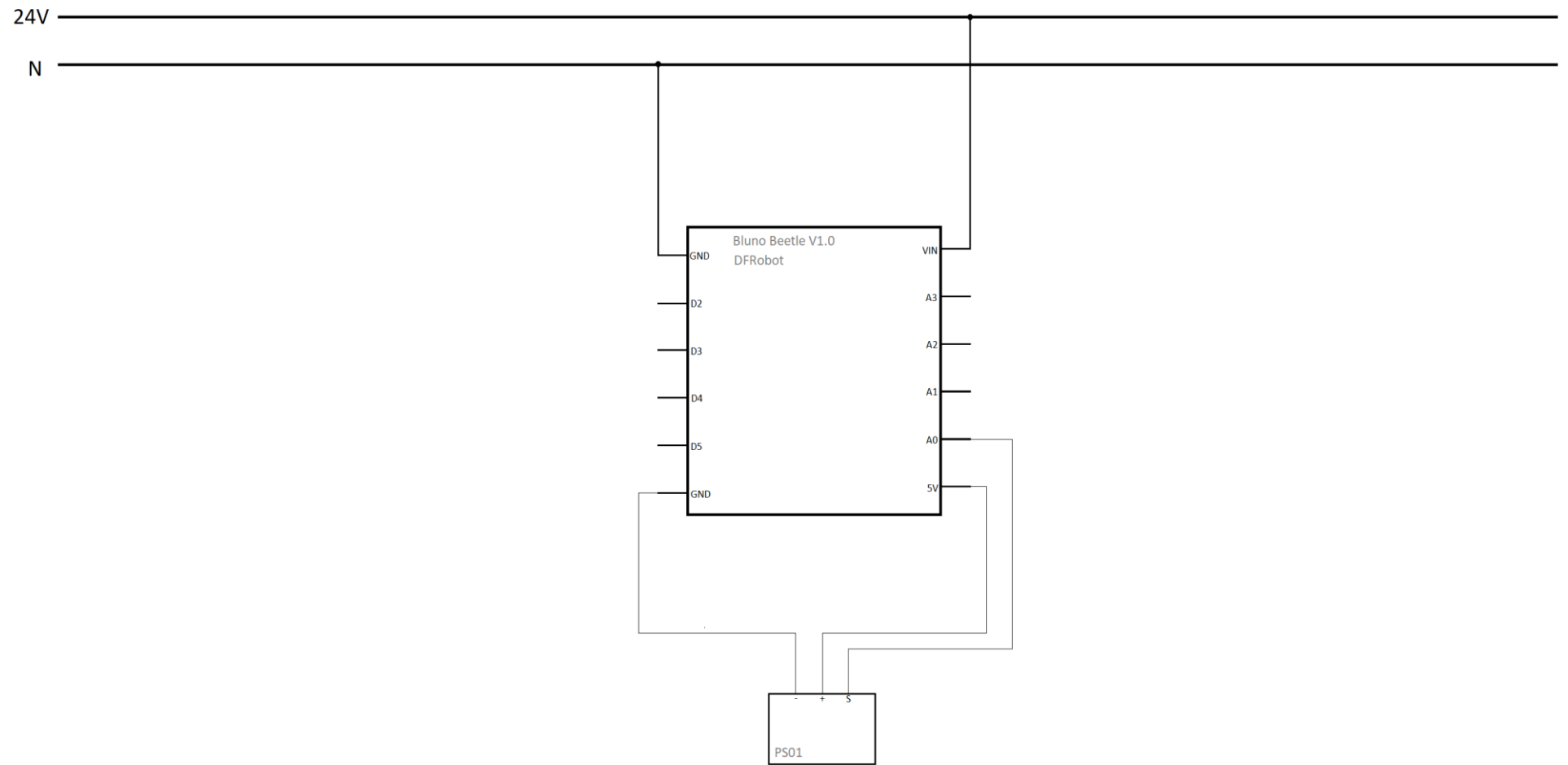


**A**---Power

**B**---Bluetooth

**C**---MCU&I/O

<b>DFROBOT</b> DRIVE THE FUTURE	
Title:	Revision: 1.0
Bluno beetle	Size: A3
File Name:	Page: 1 of 1
Drawn: DFROBOT-(CD)	Date: 2014-11-7



## 5 Testy, wyniki prac

Pierwszym krokiem było zweryfikowanie poprawności działania czujnika, czy rzeczywiście wykrywa on uderzenia serca. Czujnik wykrywa zmiany w odbitym świetle kiedy krew przepływa przez komórki, sygnał zwracany przez czujnik wygląda następująco:



Rysunek 12: przebieg wartości zwracanej przez czujnik pulsu PS01

Z naszych testów wynika, że jest to bardzo zadowalający sposób wykrywania zmian w przepływie krwi, czyli liczeniu uderzeń serca. Pulsy widoczne na wykresie są rzeczywistymi uderzeniami serca badanego.

Do naliczania uderzeń konieczny jest tzw. "poziom wyzwalaniaczyli próg powyżej, którego wartość uznawana jest za wykrycie uderzenia. Sugerowany przez producenta jest poziom 550 - zaznaczone kolorem zielonym na powyższym przebiegu (10 bitowe ADC mikrokontrolera), musieliśmy troszkę zmniejszyć, ponieważ można zauważyć, że zapas jakie mają urządzenia ponad progiem wyzwalanie jest niewielki, ustawiliśmy go na 530 - kolor czerwony.

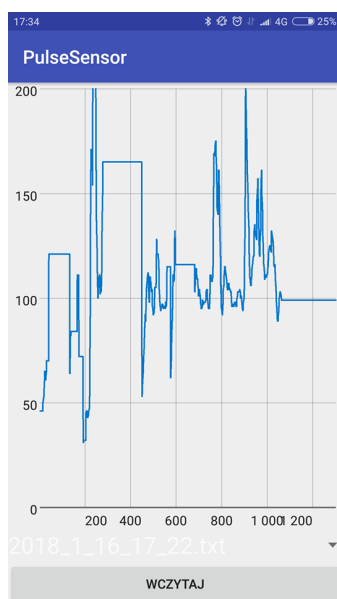
Początkowo testowaliśmy czujnik podłączając go na platformie programistycznej Arduino. Podłączając czujnik do wejścia analogowego i pobraniu odpowiedniej biblioteki do niego monitorowaliśmy jego sygnał. Sprawdzaliśmy puls mierzony przez ten czujnik oraz przez markowy ciśnieniomierz. Wyniki były zaskakująco podobne. Badaliśmy oba urządzenia w trzech różnych stanach, a pomiar wykonywaliśmy w tym samym momencie, aby wyniki były najnajbardziej wiarygodne.

Puls liczony	czujnikiem PS01	ciśnieniomierzem
Odpoczynek	83	83
	82	83
	83	84
	83	83
Po wykonaniu krótkich ćwiczeń	101	100
	101	101
	101	101
	102	100
Po wykonaniu ciężkich ćwiczeń	129	127
	128	127
	128	128
	129	128

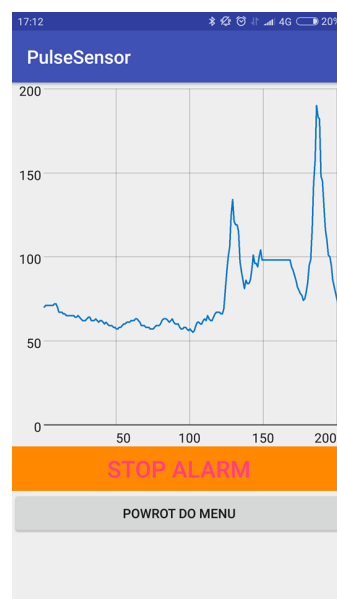
Można zauważyć, że wyniki nieznacznie się różnią. Są to niewielkie zmiany, których różnice można pominąć. Mogą być spowodowane różnicą w sposobie pomiaru samego pulsu. Ciśnieniomierz to przyrząd do pomiaru nacisku ciśnienia na powierzchnie ciał stałych, a nasz czujnik wykrywa charakterystyczne zmiany w tkankach poprzez analizę odbitego światła. Następnie po napisaniu aplikacji i ustawieniu komunikacji czujnik-smartfon, badaliśmy puls oraz działania aplikacji i budzika.

Kolejną fazą testów było sprawdzenia czy czujnik rzeczywiście wykrywa zmiany w pulsie, czujnik został przebadany na nas i rodzinie, wyniki są zadowalające. Stymulowaliśmy serce napinaniem mięśni, wstrzymywaniem oddechu itp.

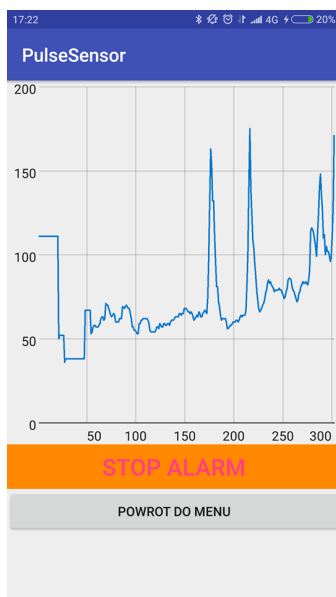




Rysunek 13: wykres pulsu osoby nr 1



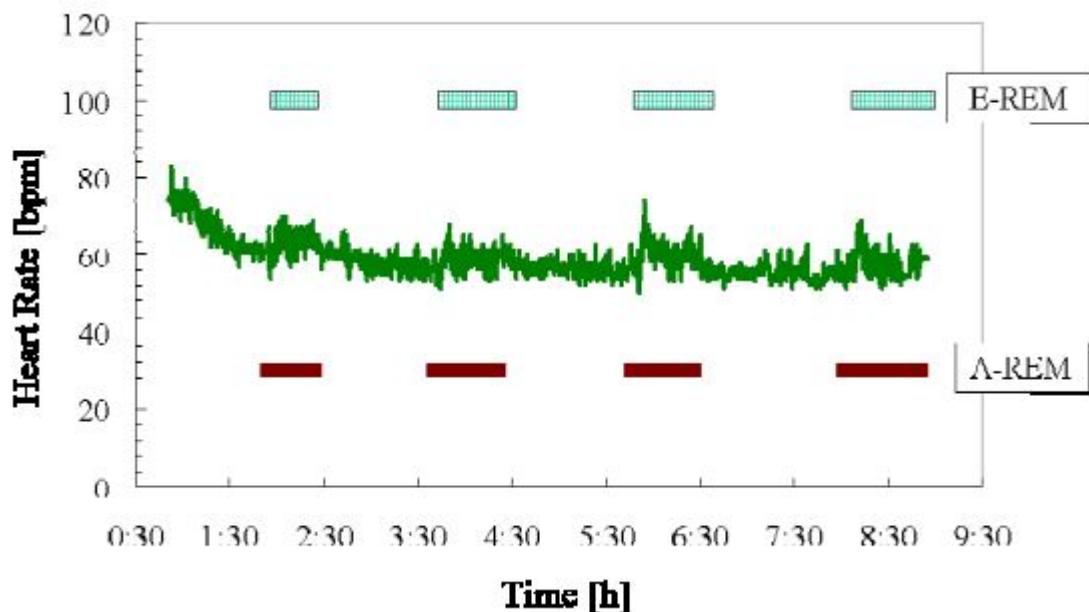
Rysunek 14: wykres pulsu osoby nr 2



Rysunek 15: wykres pulsu osoby nr 3

Przy okazji testowaliśmy skuteczność działania naszego algorytmu. Jego zasada działania opiera się o mierzenie globalnej średniej z całego pomiaru, oraz średnich tymczasowych. Globalna średnia zostaje mierzona odpowiednio wcześniej przed ustawieniem minimalnej godziny pobudki aby zebrać odpowiednio dużo próbek

aby średnia była pewna pomimo losowo napotkanych zakłóceń. Globalna średnia jest porównywana z średnimi tymczasowymi, jeżeli tymczasowa jest większa o 20 procent od średniej globalnej uznajemy za wykrycie fazy REM. Algorytm oparliśmy o wyniki badań Masao Yaso, Atsuo Nuruki, Sei-ichi Tsujimura, and Kazutomo Yunokuchi pt. "Detection of REM sleep by heart rate". Te badania było bardziej zaawansowane niż te przez nas prowadzone, detekcja fazy REM była potwierdzana przez czujniki EEG,EOG,EMG co daje praktycznie stuprocentową skuteczność w wykrywaniu. Po przeanalizowaniu tych badań i wykresów zaadoptowaliśmy algorytm na podstawie poniższego wykresu:



Rysunek 16: wykres pulsu z innych badań naukowych

Jak można zauważyć estymowane fazy REM pokrywają się prawie w 85 procentach z rzeczywistymi fazami REM, wykrytymi przez pozostałe czujniki.

## 6 Podsumowanie

Udało nam się ukończyć projekt oraz wszystkie założenia projektowe. Opaska została złożona poprawnie oraz nie ma problemów z jej działaniem i przesyłaniem sygnału wykorzystując komunikację Bluetooth. Aplikacja została stworzona z myślą o prostocie jej działania oraz łatwości w użytkowaniu, ale najważniejszym czynnikiem było zagwarantowanie użytkownikowi niezawodności w działaniu aplikacji jako budzika.

## 7 Bibliografia

- Wikipedia
- Działanie Bluetooth:
  1. <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>
  2. [https://en.wikipedia.org/wiki/Bluetooth\\_Low\\_Energy](https://en.wikipedia.org/wiki/Bluetooth_Low_Energy)
  3. <https://www.dfrobot.com/product-1259.html>
- Biblioteka do wykresów: <http://www.android-graphview.org/>
- [http://www.psychology.hes.kyushu-u.ac.jp/~lab\\_miura/Kansei/Workshop/proceedings/P-205.pdf](http://www.psychology.hes.kyushu-u.ac.jp/~lab_miura/Kansei/Workshop/proceedings/P-205.pdf)