

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH



INSTYTUT RADIOELEKTRONIKI I TECHNIK MULTIMEDIALNYCH

EAME PROJEKT

Albert Bogdanovič
Numer albumu 307870

Cel pracy

Celem mojej pracy była realizacja części hardwarowej. Zadanie polegało na wykonaniu połączenia czytnika karty SD z mikrokontrolerem Arduino nano. Dodatkowo należało dodać funkcjonalność w postaci alarmu powiadamiającego o zatrzymaniu oddechu.

W kolejnym etapie realizowałem lutowanie elementów do płytki prototypowej. Ostatecznie zbudowałem model urządzenia w postaci obudowy 3D.

Tworzenie połączenia karty SD, akcelerometru i dzwonku z uC Arduino

Realizacja zadania polegała na użyciu standardowej biblioteki Arduino do czytnika kart SD, bądź dostarczonej do akcelerometru biblioteki MPU 9250. Niestety zetknęliśmy się w trakcie realizacji z problemem niejednoznaczności adresów, gdyż adres zakupionego akcelerometru zgadzał się z adresem zegara RTC. Zamieniliśmy akcelerometr na swój własny.

Ciekawszym zadaniem była obsługa buzzera. Użyłem do tego systemu przerwań licznika Arduino nano. Co zliczenie w obsłudze jest sprawdzana wartość zmiennej counter, która zlicza czas bezruchu pacjenta. Jeżeli bezruch jest wystarczająco duży, jest uruchamiany ton na dzwonku informujący o braku oddechu. Jeżeli zaś ruch się wznowia w wystarczająco szybkim czasie – counter się resetuje.

```
cli(); //stop interrupts for till we make the settings
/*1. First we reset the control register to amke sure we start with everything disabled.*/
TCCR1A = 0; // Reset entire TCCR1A to 0
TCCR1B = 0; // Reset entire TCCR1B to 0

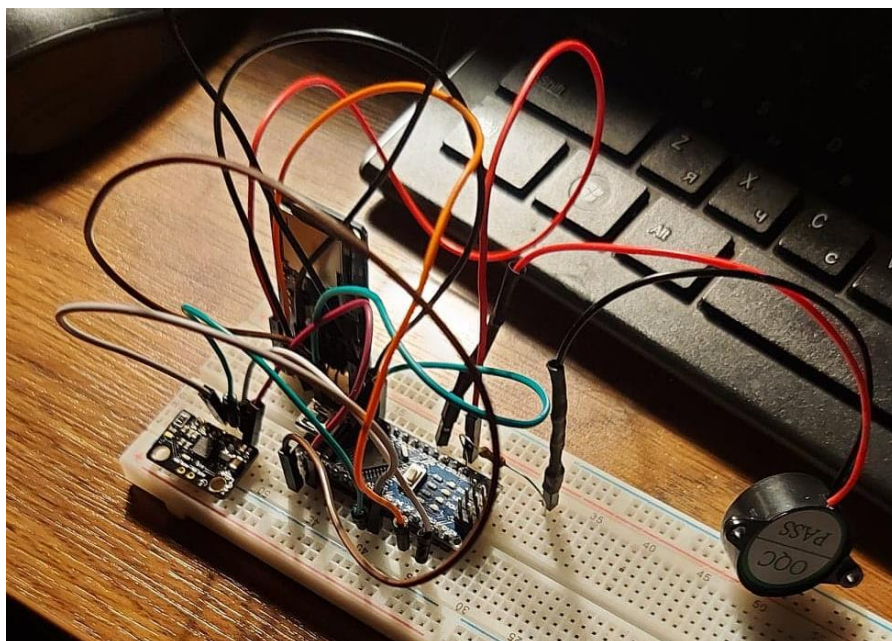
/*2. We set the prescalar to the desired value by changing the CS10 CS12 and CS12 bits. */
TCCR1B |= 00000001; // no prescalar, just clk_I0

/*3. We enable compare match mode on register A*/
TIMSK1 |= 00000010; //Set OCIE1A to 1 so we enable compare match A

/*4. Set the value of register A to 31250*/
OCR1A = 31250; //Finally we set compare register A to this value
sei(); //Enable back the interrupts

ISR(TIMER1_COMPA_vect){
  TCNT1 = 0; //First, set the timer back to 0 so it resets for next interrupt
  if(buzz_detect > 900 && buzz_detect < 950){ digitalWrite(BUZZER, HIGH); died_signal = 1000;} // Send 10KHz sound signal to 2 pin buzzer
}
```

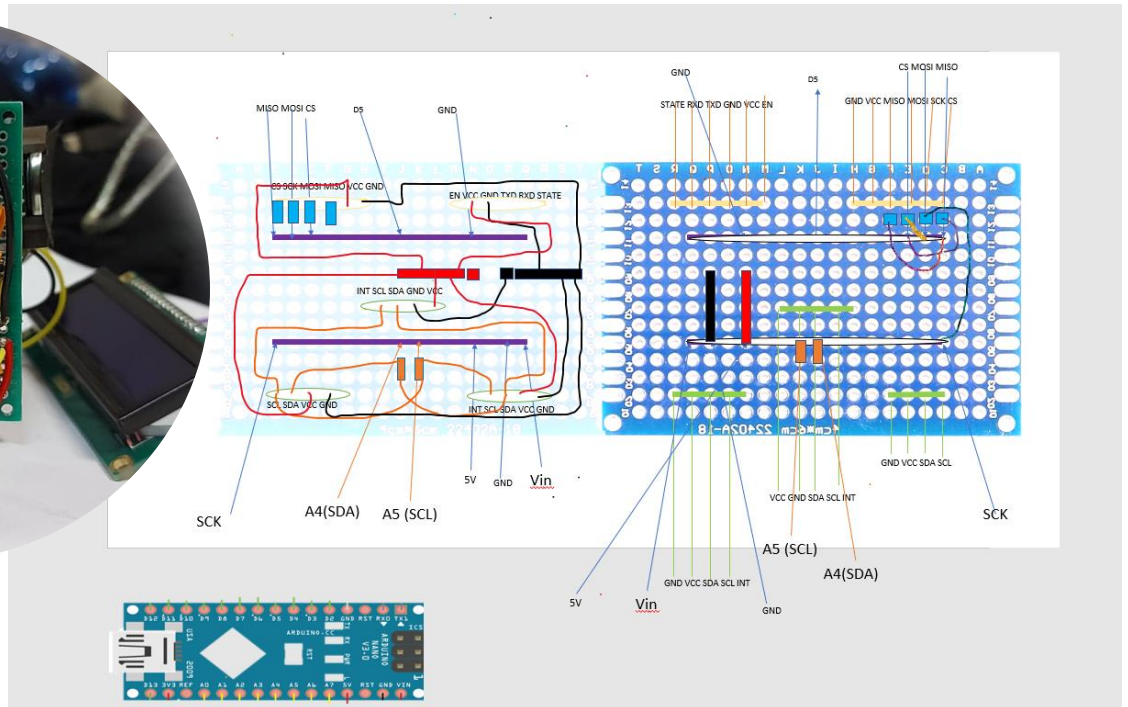
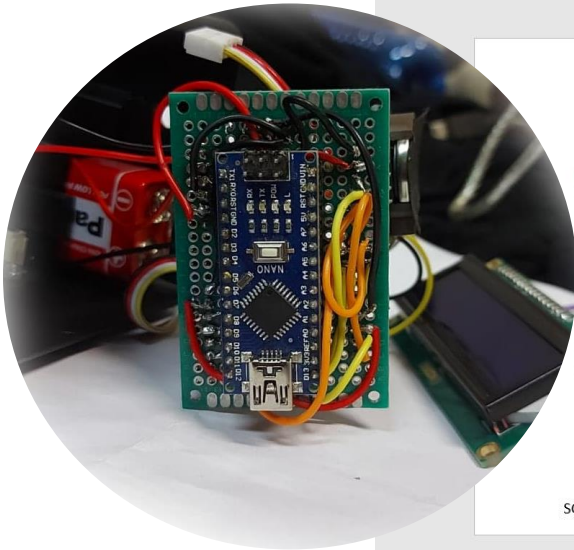
Przykład uruchomienia licznika i obsługa przerwania



Przykład zrealizowanego prototypu czytnika, akcelerometru i buzzera

Płytki prototypowa

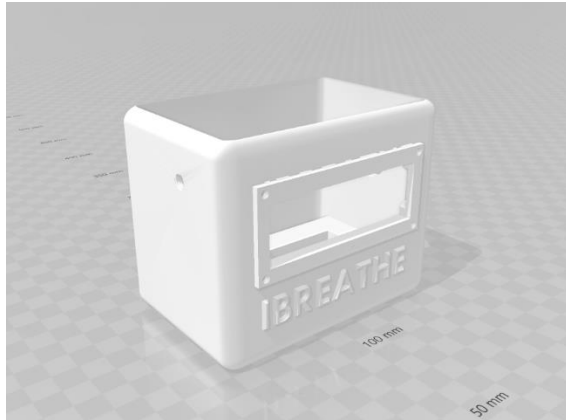
Została sporządzona płytki prototypowa. Koncepcja polega na tym, że da się z łatwością wymienić elementy modułów urządzenia. Dodatkowo taka płytki zaoszczędza miejsca, więc mogliśmy stworzyć wygodną do noszenia obudowę



*Schemat układu na płytce prototypowej i
zdjęcie płytki z umieszczonym Arduino nano*

Obudowa

Obudowa została stworzona przy wykorzystaniu programu 3D Builder. Realizacja polegała na znalezieniu open-sourcowych elementów naszego urządzenia i umieszczeniu/wycięciu ich z elementu sześciianu.



Zrzut ekranu obudowy w programie 3D Builder



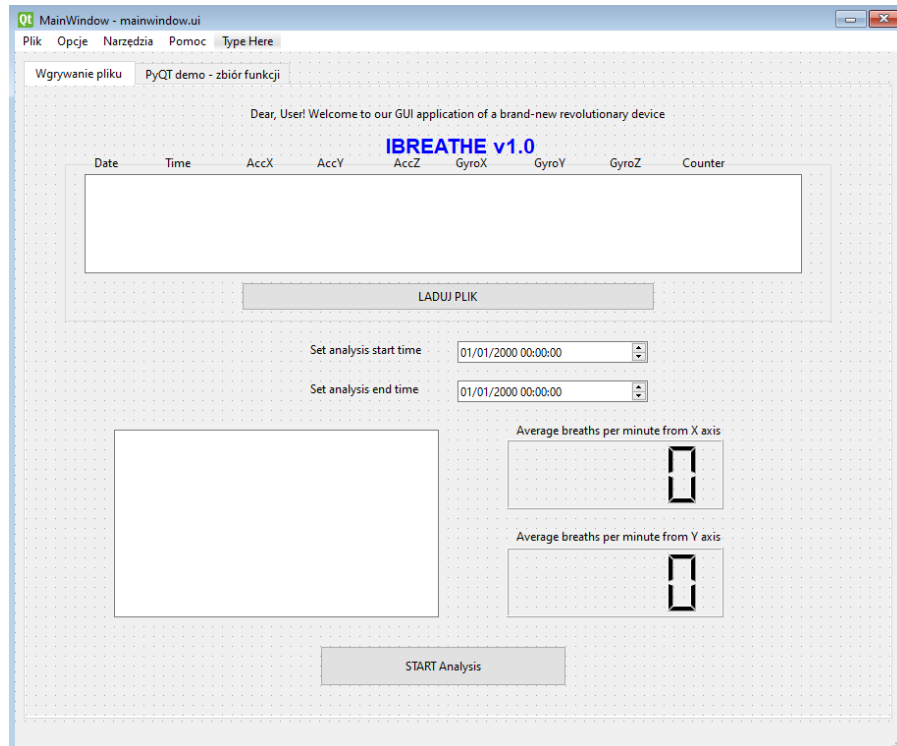
Gotowe urządzenie



Korektor postury został wykorzystany jako element do przenoszenia naszego urządzenia. Pacjent może wygodnie się poruszać, ponieważ urządzenie jest wygodnie przymocowane to korektora i pozwala nam to też monitorować oddech, ponieważ urządzenie ciągle znajduje w okolicach brzucha.

GUI

Ostatnim zrealizowanym zadaniem było połączenie części backend i frontend wykonanych przez członków zespołu. Poprawiłem IDE o dodatkowe elementy gui. Użytkownik łąduje plik txt pobrany z czytnika kart. Następnie może regulować czasem od którego zamierza analizować dane. Po naciśnięciu przycisku Start Analysis są generowane wykresy i statystyki.



Wygląd frontendu programu