

**ДВАДЕСЕТА УЧЕНИЧЕСКА КОНФЕРЕНЦИЯ**

**УК’20**

**ТЕМА НА ПРОЕКТА**

**Sleeper – анализ на качеството на съня**

**Автор:**

**Николай Златинов Стоянов**  
ППМГ „Добри Чинтулов”, гр.Сливен, 9в клас  
  
Email: **nikolay.stoyanov04@gmail.com** Телефон: **0888 883325**

**Научен ръководител (консултант):**

Петър Веселинов Стоянов

Управител, Астра Пейджинг ЕООД

Email: **peter@astrapaging.com** Телефон: **0878 624434**

**Резюме на български език  
  
Sleeper – анализ на качеството на съня**

Много хора се оплакват от трудно събуждане и хронична умора сутрин. Независимо от продължителността на съня, човек няма гаранция, че ще се събуди бодър и свеж. Основна причина за това е неспокойният и некачествен сън. Проблемът е, че никой не може да се самонаблюдава по време на сън и настоящият проект има за цел да предостави на потребителя информация за това как е спал.

Проектът се състои от хардуерно устройство, базирано на Arduino Uno и анализиращ софтуер на Python. Параметрите на съня, които системата следи, са движенията на тялото по време на сън (индикатор за неспокоен сън) и нивото на шум предизвикано от хъркане.

Проектът помага на хората да открият в кои случаи техният сън е некачествен (например след преяждане или употреба на алкохол), в кои периоди от нощта спят спокойно или неспокойно, дали хъркат през цялата нощ или само в определени периоди и други.

**Резюме на английски език  
  
Sleeper –analysis of sleep quality**

A lot of people have the problem of waking up hard and exhaustion in the morning. A long-lasting sleep doesn’t guarantee that the person will wake up fresh. This is mainly caused by low-quality sleep. The issue is that people can’t observe their own sleep and this project’s goal is to give the user information about that.

The project consists of a hardware device, based on Arduino Uno, and an analysing software coded on Python. The system keeps track of the body’s movements throughout the night (an indicator of bad sleep) and the level of sound caused by snorting.

The project helps people find the cases in which their sleep is bad (after overeating or alcohol drinking, for example) or good, whether they snort throughout the whole night or just during some periods of it.

**УВОД**Проектът има за цел да предоставя **информация за качеството на съня** на потребителя под формата на графика с три параметъра: **движение на тялото през нощта и среден и максимален шум.**    
Голям проблем, възпрепятстващ пълноценния ден на човек, са умората при събуждане и хроничната умора през целия ден. Те могат да повлияят на настроението и работоспособността на страдащия от тях и са главно предизвикани от лош и неспокоен сън, хъркане по време на сън или дори заболяване, за което дори не знаем, че ни засяга.

За жалост, няма как да се самонаблюдаваме по време на сън и тук идва на помощ sleeper. Той е устройство, на основата на Arduino Uno, което следи трите параметъра споменати в първия параграф, записва ги на SD карта в csv файл и накрая те биват визуализирани под формата на графика, с помощта на сорс код на Python.

Устройството предлага възможност за лесна промяна от потребителя, заради отворения си код и простата си изработка. То също е на значително по-ниска цена от всички подобни такива продукти. Експерименти могат да се правят и с анализиращия метод също.

Поставените цели са:

1. Информиране на потребителя за съня му.
2. Ниска цена.
3. Проста изработка.

Етапи на разработка:

1. Определяне на цялостната идея на проекта.
2. Разделяне на проекта на отделни етапи и планиране на срокове за завършване.
3. Запознаване с начина на работа на Arduino и двата основни сензора.
4. Изработване на цялостна схема на проекта и закупуване на нужните хардуерни части.
5. Създаване на цялостен сорс код, управляващ устройството и анализиращ данните
6. Тестване на устройството и събиране на база от тестове.
7. Анализиране на положителните страни и недостатъците на продукта.
8. Оформяне на документация и продължаване на развитието на проекта.

Документацията се състои от следните части:

**Част I - Проектиране и реализация на хардуера**

В тази глава съм описал основните компоненти на устройството и неговата функционална схема.

**Част II - Софтуер**

Тук е представен и разгледан подробно целия сорс код, който управлява устройството.

**Част III - Основни моменти от разработката на проекта**

В тази глава са описани основните проблеми и моменти, през които преминах в процеса на създаване на устройството.

**Част IV - Снимки и начин на работа**

Представени са снимки на крайния продукт и начина му на използване в ежедневието.

**Част V – Тестове**

Представяне на няколко теста с техните особености.

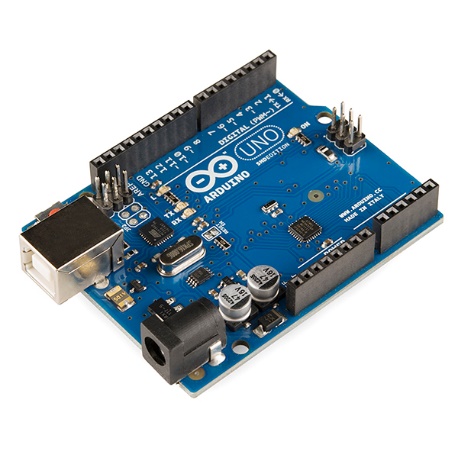
**Част VI – Заключение**

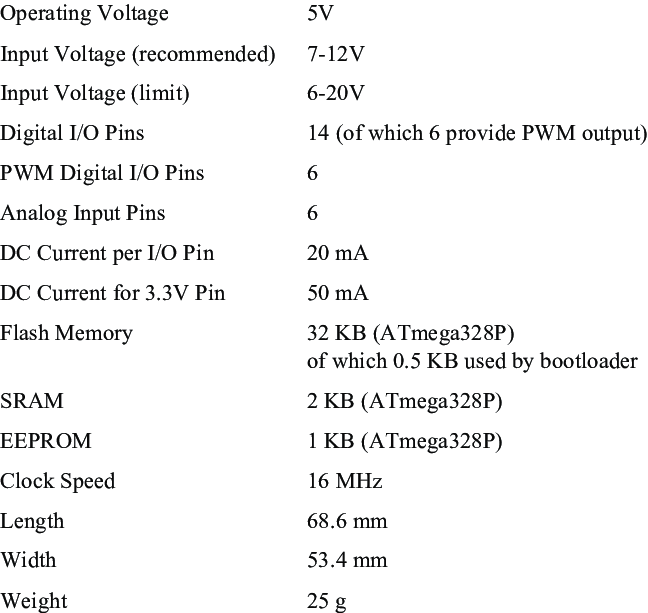
Описание на постигнатите резултати и бъдещи идеи за развитие на проекта.

**Част I - Проектиране и реализация на хардуера   
   
I.1 Компоненти**

**I.1.1 Контролер Arduino Uno**

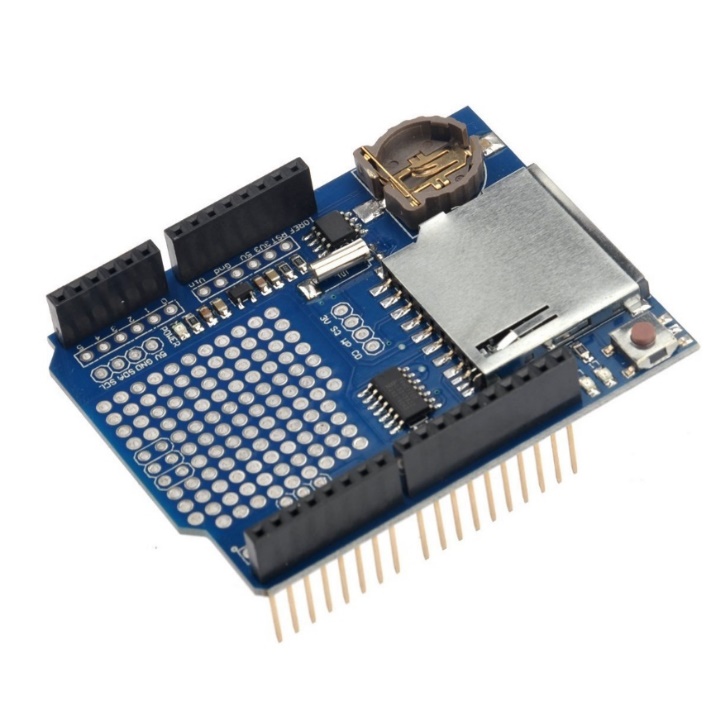
За проекта е използвано Arduino Uno, който е най-евтиният и лесен за използване хардуер. То е използвано от много начинаещи за научаването на основите в това направление на програмирането. Към него могат да се свържат много различни видове сензори и да се изобретят проекти с интересна идея и в същото време проста и евтина изработка.





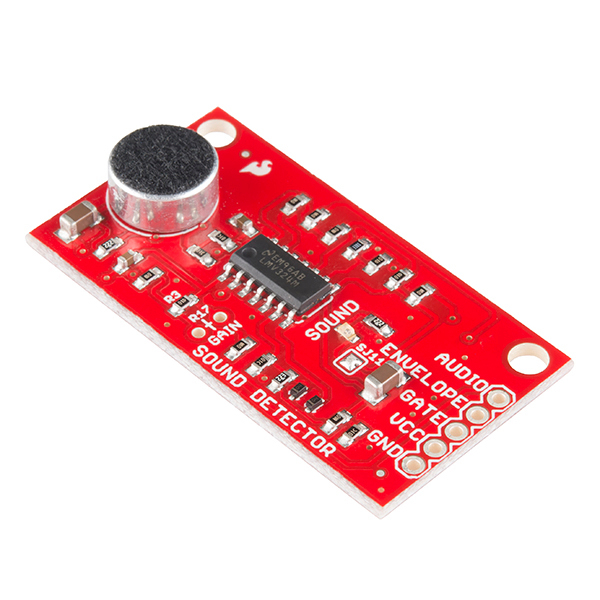
**I.1.2 Data Logger Shield V1.0**

Тази част е в основата на проекта, защото предоставя възможността за следене на времето чрез RTC (Real Time Clock) и четене и писане на SD карта. Часовникът е захранван от батерия.



**I.1.3 Микрофон на SparkFun**

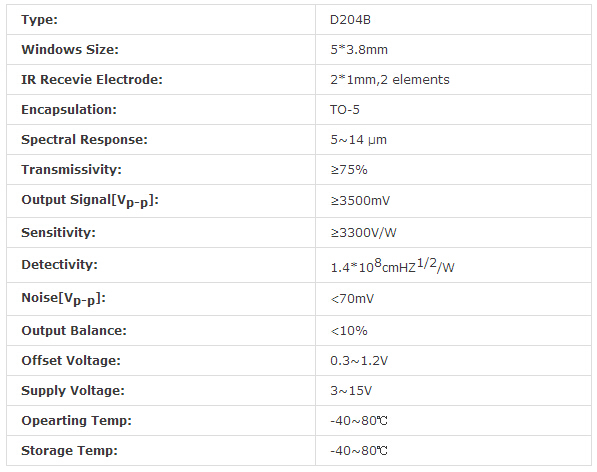
Вторият основен сензор за проекта е звуков детектор на SparkFun. Той е един от по-качествените модели микрофони, които могат да се използват с Arduino Uno. Има достатъчно добри възможности за отчитане на по-слаби звуци и три вида изход: Audio, Envelope и Gate.



**I.1.4 Инфрачервен сензор за движение HC-SR501**

Един от основните сензори на устройството е инфрачервеният сензор за движение, който измерва движенията на тялото през нощта. Той предоставя предимството да се нагласят чувствителността и времето за угасване след засечено движение. Сензорът е достатъчно добър за целите на проекта.





**I.2 Функционална схема на устройството**



**I.2.1 Описание на всички връзки между компонентите:**

* Data Logger Shield-а се поставя върху Arduino-то, като SD card logger-а заема пиновете след D10.
* Инфрачервеният сензор за движение се свързва с Arduino-то чрез пин D2.
* Звуковият сензор се свързва към Arduino микроконтролера чрез пин A0.

**Част II – Софтуер**

Софтуерът на проекта е разделен на две части. Първата е тази, която управлява устройството, отчитащ сензорите и записващ параметрите от тях заедно с датата и часа на SD картата. Той е написан на C.

Втората е частта, използвана да визуализира графиката. Тя чете записания на SD картата csv файл. Написан е на Python като са използване главно библиотеките matplotlib – за самата графика и csv – за четенето на файла.

**II.1 Сорс код управляващ устройството**

В тази част основната работа се извършва от една функция, в която има for цикъл, завъртащ се 600 пъти с 10 милисекунди delay, което сумарно прави минутата, за която се записват и след това отчитат данните. В самия цикъл параметрите се сумират в променливи, които след това се усредняват и записват в картата, чрез код в loop.

В CSV файла се записват следните стойности в посочения ред: **дата/час, движения (%), средно ниво на шум, максимално ниво на шум**. Записът се извършва един път в минута. Примерен откъс от такъв файл е показан по-долу:

**2019/11/20 23:05,4,16,3200**

**2019/11/20 23:06,0,16,3200**

**2019/11/20 23:07,20,16,3100**

Сорс кодът на устройството е показан по-долу:

#include <SdFat.h>

#include "RTClib.h"

const int chipSelect **=** 10**;** //D10 за SD card logger

const int PIR **=** 2**;** //Инициализация на пин за сензора за движение

const int MIC **=** A0**;** //Инициализация на пин за микрофона

const int number\_of\_readings **=** 600**;** //Променливите за четенията и забавянето са нагласени така,

const int func\_delay **=** 100**;** //че по-късно функцията да трае 1 минута

char file\_name**[**32**];** //Инициализация на името на файла

char date\_in\_file**[**32**];** //Инициализация на датата във файла

//Отчитане на съня

int movement**;**

int sound**;**

int max\_sound **=** 0**;**

void movements\_and\_sound\_per\_min**();** //Декларация на функцията за четене на сензорите

RTC\_DS1307 RTC**;**

SdFat sd**;**

SdFile data\_file**;**

DateTime now**;**

void setup**(**void**)** **{**

Serial**.**begin**(**9600**);**

RTC**.**begin**();**

**if(!**RTC**.**isrunning**()){**

Serial**.**println**(**"RTC is NOT running!"**);**

RTC**.**adjust**(**DateTime**(**F**(**\_\_DATE\_\_**),** F**(**\_\_TIME\_\_**)));**

**}**

**if(!**sd**.**begin**(**chipSelect**,** SPI\_HALF\_SPEED**))** **{** Serial**.**println**(**"SD card failed or not present"**);** **}**

now **=** RTC**.**now**();**

sprintf**(**file\_name**,** "%4d%02d%02d%02d%02d.csv"**,**now**.**year**(),** now**.**month**(),** now**.**day**(),** now**.**hour**(),** now**.**minute**());** //Създаване на името на файла

**}**

void loop**()** **{**

Serial**.**println**(**"loop"**);**

now **=** RTC**.**now**();**

sprintf**(**date\_in\_file**,** "%4d/%02d/%02d %02d:%02d"**,** now**.**year**(),** now**.**month**(),** now**.**day**(),** now**.**hour**(),** now**.**minute**());** //Създаване на датата във файла

movements\_and\_sound\_per\_min**();**

//Отваряне и принтиране във файла

Serial**.**println**(**"file open"**);**

**if(**data\_file**.**open**(**file\_name**,** O\_RDWR **|** O\_CREAT **|** O\_AT\_END**)){**

data\_file**.**print**(**date\_in\_file**);**

data\_file**.**print**(**","**);**

data\_file**.**print**(**movement**);**

data\_file**.**print**(**","**);**

data\_file**.**print**(**sound**);**

data\_file**.**print**(**","**);**

data\_file**.**println**(**max\_sound**);**

data\_file**.**close**();**

Serial**.**println**(**"file closed"**);**

**}**

**}**

void movements\_and\_sound\_per\_min**(){**

//Променливи за движението

int present\_reading\_movement **=** 0**;**

float sum\_of\_readings\_movement **=** 0**;**

movement **=** 0**;**

//Променливи за звука

int present\_reading\_sound **=** 0**;**

int past\_reading\_sound **=** 0**;**

float sum\_of\_readings\_sound **=** 0**;**

int deviation **=** 0**;**

max\_sound **=** 0**;**

sound **=** 0**;**

**for(**int i**=**0**;** i**<**number\_of\_readings**;** i**++){**

//Отчитане на движение

present\_reading\_movement **=** digitalRead**(**PIR**);**

sum\_of\_readings\_movement **+=** present\_reading\_movement**;**

//Отчитане на звук

present\_reading\_sound **=** analogRead**(**MIC**);**

sum\_of\_readings\_sound **+=** present\_reading\_sound**;**

**if(**max\_sound **==** 0 **||** present\_reading\_sound **>** max\_sound**)** max\_sound **=** present\_reading\_sound**;**

delay**(**func\_delay**);**

**}**

movement **=** sum\_of\_readings\_movement **/** number\_of\_readings **\*** 100**;**

sound **=** sum\_of\_readings\_sound **/** number\_of\_readings **\*** 100**;**

max\_sound **=** max\_sound **\*** 100**;**

**}**

**II.2 Сорс код на анализиращия софтуер**

Тук кодът е съставен основно от функции от Python библиотеката matplotlib, за създаване на графиката. Има и допълнителни редове за стилистичното изграждане на графиката и осигуряване на четимостта й.

**import** matplotlib**.**pyplot **as** plt

**import** matplotlib**.**dates **as** mdates

**import** numpy **as** np

**import** pandas **as** pd

**import** csv

x **=** **[]**

y1 **=** **[]**

y2 **=** **[]**

y3 **=** **[]**

**with** open**(**'Niki2.CSV'**,**'r'**)** **as** csvfile**:**

plots **=** csv**.**reader**(**csvfile**,** delimiter**=**','**)**

**for** row **in** plots**:**

x**.**append**(**pd**.**to\_datetime**(**row**[**0**]))**

y1**.**append**(**float**(**row**[**1**]))**

y2**.**append**(**float**(**row**[**2**]))**

y3**.**append**(**float**(**row**[**3**])** **/** 100**)**

fig **=** plt**.**figure**(**figsize**=(**25**,**10**))**

ax **=** fig**.**add\_subplot**()**

ax**.**set\_title**(**'Графика на съня'**,** fontsize**=**50**,** pad**=**25**,** fontweight**=**'bold'**)**

ax**.**set\_xlabel**(**'Нива на параметрите'**,** fontsize**=**30**,** labelpad**=**20**)**

ax**.**set\_ylabel**(**'Време'**,** fontsize**=**30**,** labelpad**=**20**)**

ax**.**plot**(**x**,** y1**,** label**=**'Движение'**,** color**=**'tab:green'**)**

ax**.**plot**(**x**,** y2**,** label**=**'Среден шум'**)**

ax**.**plot**(**x**,** y3**,** label**=**'Най-висок шум'**)**

ax**.**xaxis**.**set\_major\_locator**(**matplotlib**.**dates**.**MinuteLocator**(**interval**=**30**))**

ax**.**xaxis**.**set\_major\_formatter**(**mdates**.**DateFormatter**(**'%H:%M'**))**

**for** tick **in** ax**.**xaxis**.**get\_major\_ticks**():**

tick**.**label**.**set\_fontsize**(**14**)**

**for** tick **in** ax**.**yaxis**.**get\_major\_ticks**():**

tick**.**label**.**set\_fontsize**(**14**)**

ax**.**legend**(**loc**=**9**,**fontsize**=**14**)**

fig**.**savefig**(**"example3.png"**)**

**Част III - Основни моменти от разработката на проекта**   
За постигането на завършеност и функционалност трябваше да се мине през определени стъпки и да се изгладят някои проблеми по време на разработката.

Първата стъпка беше съставянето на алгоритъм, отговарящ на целите, за отчитане на сензорите и записване на картата през подходящо време. Проблемът съпътстван с това беше осигуряване на максимален живот на картата. Решението беше, създаване на софтуер, отчитащ сензорите 10 пъти в секунда, натрупващ данните от 1 минута и записващ всичко това наведнъж в SD картата.

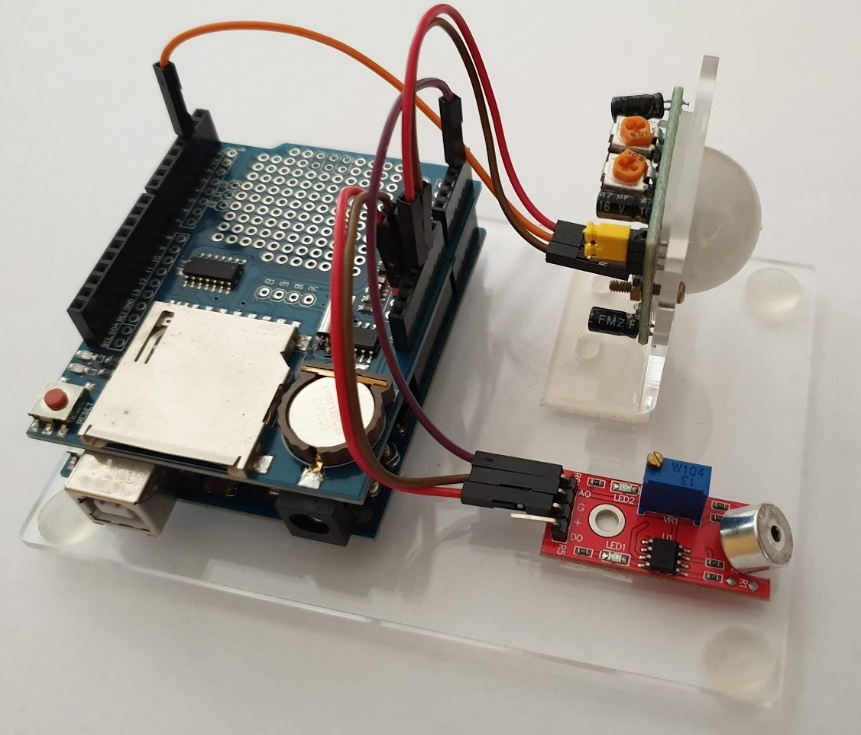
Следващата стъпка беше именуването на създадения файл. За да се разграничават различните файлове максимално добре, реших името му да е годината, датата и часа на създаването му. Доколкото вида файл, се спрях на csv файл, за по-лесна нататъчна обработка чрез Python.

След сглобяването на хардуера и осъществяването на анализиращия софтуер следваха тестовете. За които беше нужен по-дълъг период от време и събиране на по-интригуващи тестове от различни хора.

**Част IV - Снимки и начин на работа**

**IV.1 Завършеното устройство**

Готовото устройство е много лесно за употреба. Преди лягане е необходимо да се сложи на нощното шкафче до леглото и да се включи към контакт с адаптер за USB (например от мобилен телефон).



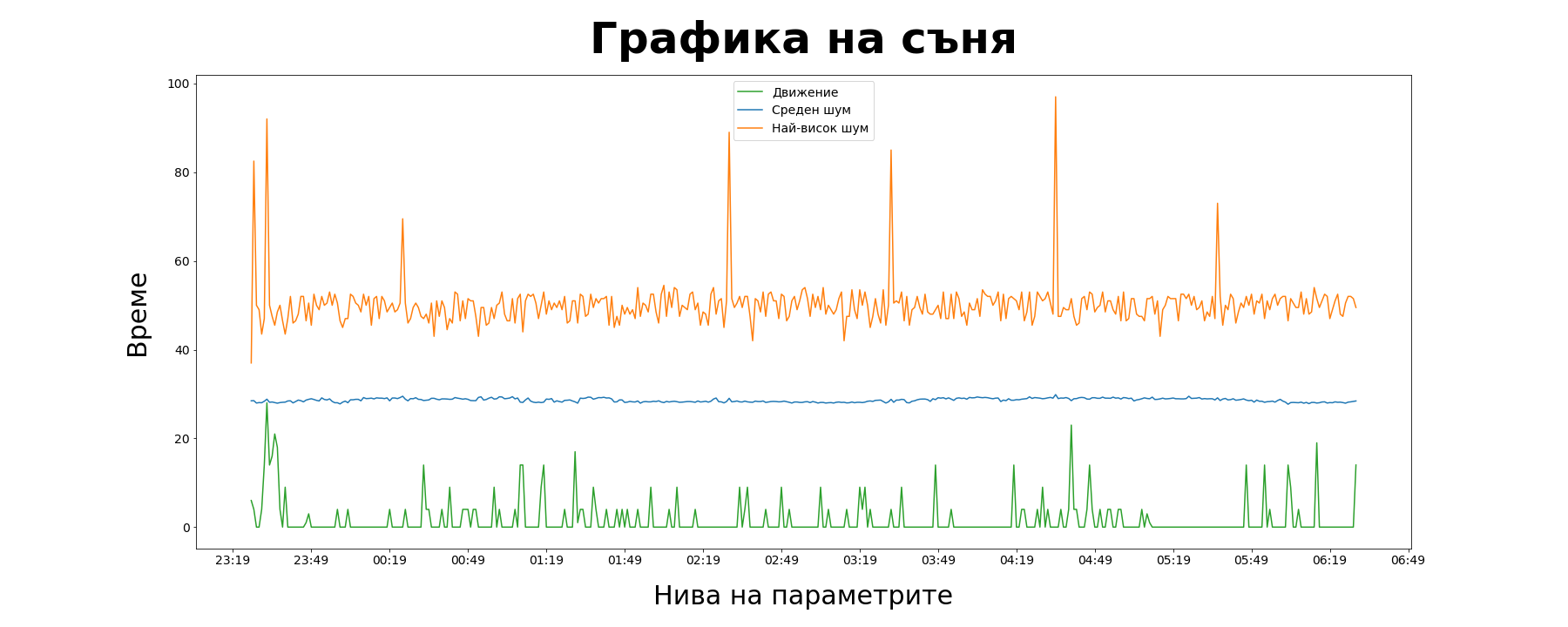
**IV.2 Начин на работа:**

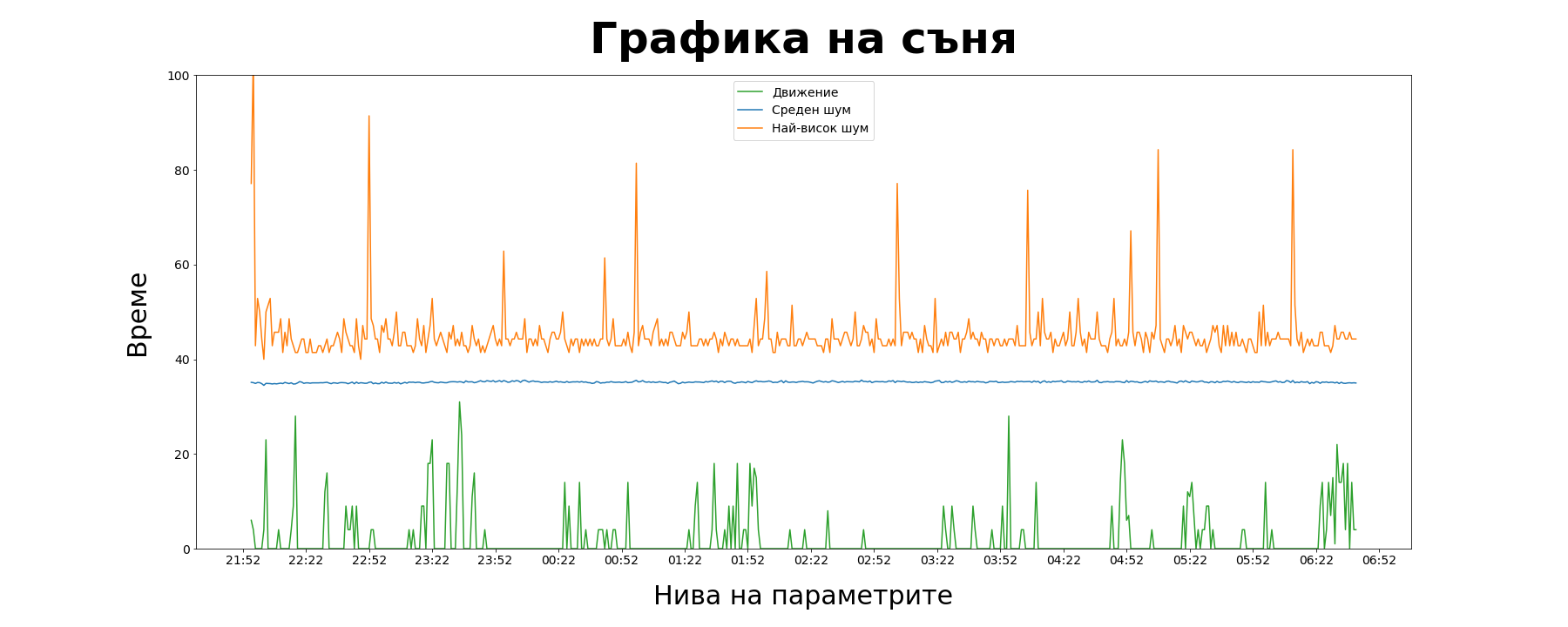
При включването си устройството започва да отчита параметрите: движения на тялото и максимален и среден шум**.** То натрупва тези данни в рамките на 1 минута, след което те биват усреднявани за записани на SD картата, заедно с датата и часа им на записване.

След като данните от цялата нощ бъдат събрани в csv файл, той бива обработен чрез софтуерът на Python, който ги превръща в графика, лесна за анализиране.

**Част V - Тестове**

Представени са 2 графики, в които се вижда разликата в съня на определен човек през две различни вечери. Първият човек има доста по-краткотрайни движения, но заспива по-трудно, докато при втория са по-чести и на периоди и той се събужда по-трудно. Веднага можем да забележим и по-шумната обстановка на втората графика, благодарение на най-високия шум.





По този начин могат да се изследват резултатите предоставени от проекта. С изследване на сънните навици на един човек в продължение на по-дълъг период от време и в различни ситуации (приемане на алкохол или по-ранно лягане) и условия (шумна обстановка или краткотраен, но силен шум). С тази информация човек може да подобри съня си, според нуждите си и да придобие по-добра цялостна представа за себе си.

**Част VI – Заключение**

**VI.1 Цел и постигане на целта**

Целта на проекта е да помогне на хората да научат повече за собствения си сън. Това се постига чрез устройство за следене на параметрите на съняреализирано с Arduino Uno със софтуер написан на C, отчитащ два основни сензора: за движение и микрофон и записващ данните в SD карта. Информацията събрана бива визуализирана, за по-лесна обработка, от сорс код, написан на Python.   
Основните предимства на устройството и софтуера са:

* ниска цена и лесно за изработка
* отворен код, позволяващ на всеки да експериментира с различни методи за анализ на данните
* широки възможности за добавяне на нови сензори

**VI.2 Използвани софтуерни продукти**

За реализацията на проекта са използвани безплатните средите за разработка – Arduino IDE и Jupyter Notebook.

**VI.3 Бъдещо развитие**

Плановете за развитие на проекта включват:

* Добавяне на трета величина за измерване и изследване – пулс, която би дала възможност за откриване на сериозни проблеми с дишането водещи до рязко повишаване на пулса (напр. сънна апнея).
* Създаване на web сайт, в който всеки анонимно да може да качва своите резултати и да получава, както анализи за изминалата нощ, така и тенденции в по-продължителен период.
* Набиране на повече тестове, отчетени от различни хора и от един човек в продължение на повече време. Използване на различни анализиращи методи.

**ИЗТОЧНИЦИ НА ИНФОРМАЦИЯ**

* Работа с PIR Motion sensor – <https://create.arduino.cc/projecthub/electropeak/pir-motion-sensor-how-to-use-pirs-w-arduino-raspberry-pi-18d7fa>
* Работа с микрофон - <http://arduinolearning.com/code/ky038-microphone-module-and-arduino-example.php>
* Библиотека matplotlib - <https://matplotlib.org/>