

**Автономная некоммерческая общеобразовательная организация
"Физтех-лицей"
(АНОО «Физтех-лицей» им. П.Л. Капицы)**

XX научно-практическая конференция

«Старт в инновации»

Устройство для контроля осанки

Выполнила:
Константинова Варвара 10В
Руководитель:
Зыонг К. К.

Московская область, г. Долгопрудный

2021 г.

Введение

Многие даже не догадываются, что хорошая осанка так же важна, как здоровая диета, комфортный сон и физические упражнения. Она гарантирует, что тело хорошо подготовлено к выполнению ежедневных задач с большей энергией, все время сохраняя бодрость. Это основа общего физического здоровья.

Поддерживать хорошую осанку довольно легко, однако не все люди способны сами заметить за собой, что сидят с согнутой спиной.

Формирование осанки происходит в течение всей жизни человека, однако в детском и подростковом возрасте закладываются ее главные характеристики. В следствие этого осанка без патологий является важным фактором полноценного развития подрастающего поколения. В случае если не заниматься данной проблемой в подростковом возрасте, в дальнейшем имеют все шансы возникнуть нарушения, которые можно будет исправить лишь хирургическим методом.

В настоящее время учебный процесс протекает при доминирующей сидячей позе, что может вызвать изменение осанки в худшую сторону. Во избежание этого следует проводить мероприятия, которые будут направлены на профилактику, а также коррекцию нарушений опорно-двигательного аппарата. К таким мероприятиям можно отнести контроль позы учащихся во время занятий и выполнения домашней работы.

Для упрощения контроля осанки было принято решение создать устройство под названием Posture Checker на базе платформы Arduino в комплекте с приложением на Android, разработанным в MIT App Inventor.

Цель работы

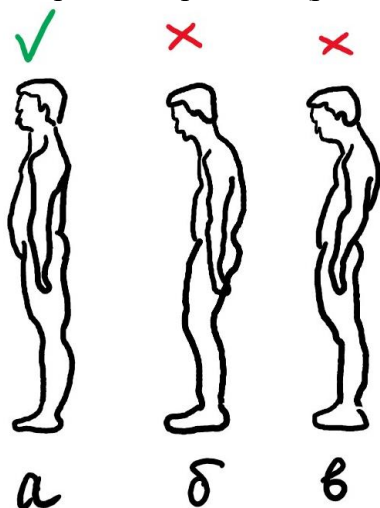
Создание устройства для контроля осанки, разработка приложения для управления им.

Задачи:

1. Проанализировать медицинские данные и выявить нужные для определения правильности осанки параметры
2. Собрать устройство, запрограммировать его
3. Создать приложение для смартфона
4. Оформить инструкции для работы с устройством

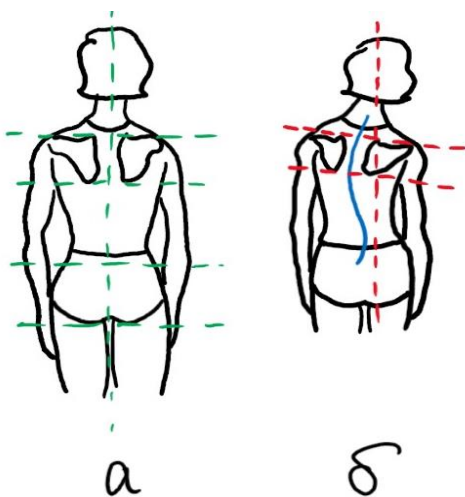
Последствия нарушения осанки [1]

Осанка – это привычная поза, непринужденно стоящего человека, которую он принимает без излишнего мышечного напряжения. Хорошая осанка означает, что все наше тело симметрично выровнено (рис.1.а, рис.2.а).



Сутулость — это увеличение грудного кифоза с одновременным уменьшением поясничного лордоза. Данная патология связана с нарушением осанки в сагиттальной плоскости, вызывающая увеличение одного или нескольких физиологических изгибов, так и с уменьшением их (рис.1.б, 1.в).

Рис. 1 Нарушение осанки в сагиттальной плоскости



Возникновение изгиба позвоночника во фронтальной плоскости называется сколиотической осанкой. Если возникает такое нарушение, то, как следствие возникает асимметрия туловища, которая проявляется в разной высоте плеч, положении лопаток по отношению к позвоночнику, к грудной стенке (рис.2.б).

Рис. 2 Нарушение осанки во фронтальной плоскости

Нарушение осанки должно вызывать особое беспокойство, так как оно часто приводит к очень заметному сбою в работе внутренних органов. Одним из самых серьёзных последствий, угрожающих жизни человека, можно отметить нарушение работы сердца. Нельзя не упомянуть воздействие неправильной осанки в виде уменьшения объёма лёгких, замедления метаболизма, затруднения работы желудочно-кишечного тракта. Перечень проблем продолжают головные боли, повышенная утомляемость и плохой аппетит. Таким образом, человек с нарушением осанки становится вялым, апатичным, менее активным. Появление остеохондроза, является тоже следствием неправильной осанки.

Признаки правильной осанки [3]

- равномерно выраженные физиологические изгибы позвоночника в сагиттальной плоскости;
- наличие физиологических изгибов в сагиттальной (проходит посередине тела, делит его на две симметричные половины, правую и левую) плоскости в виде шейного и поясничного лордоза и кифоза в грудном и крестцовых отделах;
- прямая конфигурация позвоночного столба во фронтальной плоскости (при осмотре со стороны спины);
- глубина лордоза в шейном и поясничном отделах позвоночника соответствует толщине ладони обследуемого пациента.

Описание устройства

Устройство представляет собой систему из двух датчиков GY-521 (гироскоп + акселерометр), расположенных на спине в районе шеи (рис.3.1) и талии (рис.3.2), подключенных к устройству на плате Arduino Nano. Датчики крепятся к одежде человека максимально близко к телу, а корпус устройства на пояс.

Для начала работы устройство подключается к телефону с приложением при помощи Bluetooth. Там же пользователь устанавливает параметры: правильная осанка/неправильная осанка. Исходя из них, устройство в дальнейшем будет определять, правильно ли сидит человек. Параметры можно изменить в любой момент. Важно: после выключения процедуру стартовой настройки нужно повторять заново.

Когда осанка не соответствует правильной, устройство подает звуковой и вибро сигналы. Как только человек принимает правильное положение, сигнал прекращается. Состояние осанки проверяется каждую секунду, чтобы избежать случайных (ложных) "сигнализаций".

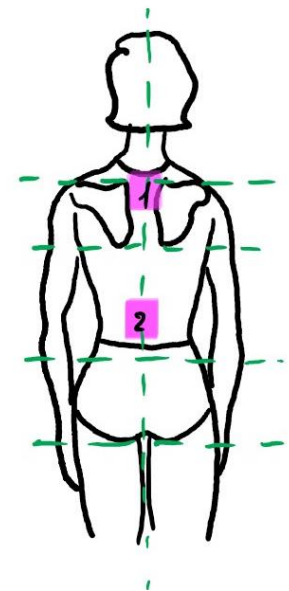


Рис. 3 Расположение датчиков

Принцип работы

Рассмотрим прошивку устройства. Ссылка на неё есть в пункте 1 приложения 4.

Используя показания двух датчиков GY-521, получаем значения по двум осям координат (строки кода 59-60). Далее они проходят процесс проверки.

```
57 void loop() {  
58  
59   MPU_d.Execute(); //в цикле всегда опрашиваем нижний датчик  
60   MPU_u.Execute(); //в цикле всегда опрашиваем верхний датчик  
61 }
```

Для неё используются значения, записанные лично пользователем как хорошая и плохая осанка (строки кода 84-101).

```

63 // получение данных с андроида
64
65 if (btSerial.available()) { //если с приложения что-то пришло
66     while (btSerial.available()) { //пока в буфере bt что-то есть
67
68         mode = btSerial.parseInt(); // цифровая команда с андроида
69
70         if (btSerial.read() == '\n') { //конец команды
71
72             if(mode ==0 ){ //получили '0'
73                 //режим в котором ждем, когда возникнет искривление, и ругаемся
74                 Serial.println("MODE: Normal");
75                 notification(1,1);
76             }
77
78             if(mode ==2 ){ //получили '2'
79                 //Debug info отправляем оси в BT
80                 Serial.println("MODE: Debugging");
81                 notification(1,1);
82             }
83
84             if(mode ==3 ){ //получили '3'
85                 //все хорошо
86                 x1_normal = mpu_d.GetAngX();
87                 y1_normal = mpu_d.GetAngY();
88                 x2_normal = mpu_u.GetAngX();
89                 y2_normal = mpu_u.GetAngY();
90                 notification(1,1);
91                 // ----- тут запоминаем "хорошее" положение
92             }
93
94             if(mode ==4 ){ //получили '4'
95                 //все плохо
96                 x1_bad = mpu_d.GetAngX();
97                 y1_bad = mpu_d.GetAngY();
98                 x2_bad = mpu_u.GetAngX();
99                 y2_bad = mpu_u.GetAngY();
100                 notification(1,1);
101             }

```

Ниже приведена программа, которая легла в основу проверки положения человека. Ссылка на неё есть в пункте 2 приложения 4.

```

22 #include <iostream>
23 using namespace std;
24
25 float x1, y1, x2, y2, a1, b1, a2, b2;
26
27 bool intersection (float x1, float y1, float x2, float y2, float
a1, float b1, float a2, float b2){
28
29     return !( (y1 + a1) < (y2 - a2) || (y1 - a1) > (y2 + a2) ||
(x1 + b1) < (x2 - b2) || (x1 - b1) > (x2 + b2) );
30     // проверяем от противного
31
32 }

```

Эта программа решает задачу, поставленную так:

```

2  даны два прямоугольника с центрами в точках (x1; y1) и (x2; y2),
3  ширина и высота первого соответственно 2*a1 и 2*b1,
4  ширина и высота второго соответственно 2*a2 и 2*b2.
5
6  вывести 1, если прямоугольники пересекаются/содержатся один в
   другом / касаются
7  иначе вывести 0.
8  _____
9
10 (x1 - b1; y1 + a1) - верхний левый угол
11 .....
12 |                     |
13 |                     |
14 |                     |
15 |                     |
16 | | | | | | | (x1 + b1; y1 - a1) -нижний правый угол
17 |
18 аналогично для второго прямоугольника
19 */

```

В прошивке устройства всё работает так же, но проверка возвращает результаты “наоборот”.

```

24 bool intersection (double x1, double y1, double x2, double y2, int a1, int b1, int a2, int b2){
25     return ( (y1 + a1) < (y2 - a2) || (y1 - a1) > (y2 + a2) || (x1 + b1) < (x2 - b2) || (x1 - b1) > (x2 + b2) );
26     // проверяем от противного,если нет пересечения, возвращает 1, если есть 0
27 }

```

Обработку данных для получения значений a1, b1, a2 и b2 можно представить в виде таблицы.

Нахождение параметров прямоугольников			
current	normal	bad	delta
x1	x1_n	x1_b	a1 = x1_n - x1_b
y1	y1_n	y1_b	b1 = y1_n - y1_b
x2	x2_n	x2_b	a2 = x2_n - x2_b
y2	y2_n	y2_b	b2 = y2_n - y2_b

В программе это выглядит следующим образом:

```

if(mode == 5){ // проверяем осанку каждые DEBUG_SEND_SPEED если mode=5
    a1 = abs(x1_normal - x1_bad) + 10;
    a2 = abs(x2_normal - x2_bad) + 10;
    b1 = abs(y1_normal - y1_bad) + 3;
    b2 = abs(y2_normal - y2_bad) + 3;

    if(intersection(mpu_d.GetAngX(), mpu_d.GetAngY(), mpu_u.GetAngX(), mpu_u.GetAngY(), a1, b1, a2, b2)){
        if(flag_of_bad_posture == 1){
            btSerial.println("ALARM~");
            notification(1,1);
        }
        else flag_of_bad_posture = 1;
    }
    else flag_of_bad_posture = 0;
}

```

Дополнительно есть флаг (flag_of_bad_posture), показывающий, находился ли человек в неправильном положении во время предыдущей проверки. Это нужно, чтобы по максимуму избавиться от случайных уведомлений.

Материалы и оборудование

- Плата Arduino Nano
- Bluetooth модуль JDY-31
- Гироскоп + акселерометр GY-521 (x2)
- Модуль заряда аккумуляторов (с защитой) TP4056 micro USB
- DC/DC преобразователь MT3608
- Литиевый аккумулятор из старого телефона
- Вибромотор из старого телефона
- Пьезоэлемент (“пищалка”)
- Текстолитовая макетная плата под пайку
- Провода монтажные
- Корпус пластиковый для устройства
- Корпус пластиковый для датчика (x2)
- Паяльник

Процесс изготовления

Сначала было проведено микроисследование о принципе работы с датчиками GY-521 (представляют собой 3-х осевой гироскоп и 3-х осевой акселерометр в одном корпусе). Гироскоп измеряет угловую скорость вращения вокруг оси, условно в градусах/секунду. Для нахождения текущего угла по скорости нужно интегрировать эту скорость. Акселерометр измеряет ускорение вдоль оси, условно в метрах / секунду². Если датчик лежит на столе или движется с постоянной скоростью, на оси будет спроецирован вектор силы тяжести (по сути вектор ускорения свободного падения). Если датчик движется с ускорением, вдобавок к ускорению свободного падения получим составляющую вектора ускорения. Если датчик находится в свободном падении, величины по всем осям будут нулевыми. Зная проекции вектора силы тяжести можно с достаточно высокой точностью определить угол наклона датчика относительно этого вектора.

По отдельности гироскоп и акселерометр не могут обеспечить точный контроль положения датчика, но если совмещать их показания, картина получается вполне точной.

Датчики подключаются на шину i2c (**SDA -> A4, SCL -> A5, GND -> GND**). На платах датчиков стоит стабилизатор, позволяющий питаться от пина 5V (**VCC -> 5V**). На модуле

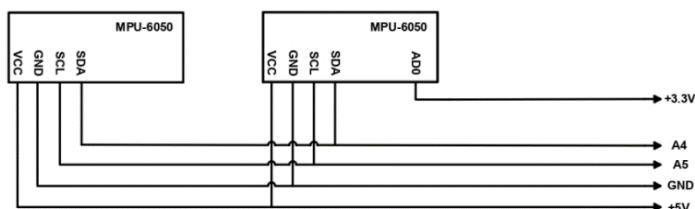


Рис. 4 Подключение датчиков к ардуино

есть пин **AD0**. Если он никуда не подключен (или подключен к **GND**, т.е. имеет на входе логический '0') – адрес датчика на шине i2c будет 0x68; если имеет на входе логическую '1' что для данного устройства эквивалентно уровню питания - 3,3 вольта (**VCC**) – адрес будет 0x69. Таким образом, без дополнительных микросхем можно подключить два датчика с разными адресами. [2]

После подключения датчиков к ардуино была обнаружена проблема – ненадежные и очень тонкие провода, которым ни в коем случае обрываться нельзя. Было решено напечатать на 3D-принтере корпуса для датчиков, чтобы обеспечить датчикам защиту (см. приложение 5).

Затем были собраны элементы, необходимые для возможности заряжать аккумулятор и пользоваться устройством без подключения к электричеству.

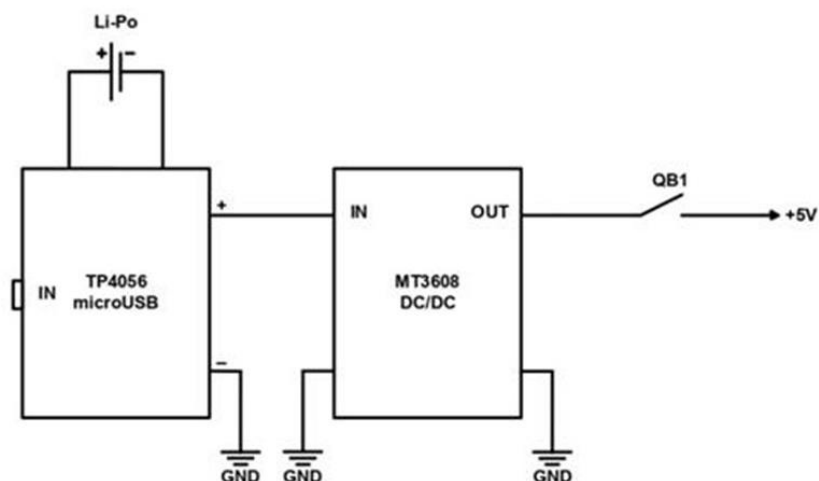


Рис.5 Устройство аккумулятора

Далее было организовано Bluetooth подключение между телефоном и устройством, а также при помощи MIT App Inventor был создан черновой вариант приложения.

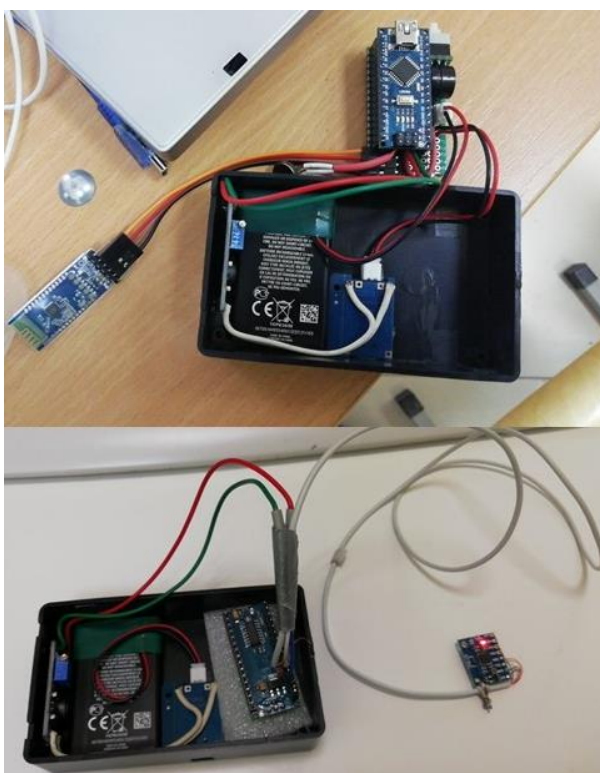


Рис. 6 Подключение Bluetooth



Рис. 7 Пробная версия приложения

В дальнейшем все модификации устройства были незначительными – был добавлен пьезоэлемент и вибромотор, подключение датчиков «переехало» на текстолитовую плату. Итоговый вид снаружи и изнутри можно увидеть соответственно в приложениях 1б и 2.

Приложение было переработано: изменился дизайн, была добавлена краткая инструкция по использованию. Интерфейс приведён в приложении 3а, блоки кода – в приложении 3б.

После завершения работы, было проведено несколько экспериментов, по итогу которых устройство показало удовлетворительные результаты.

Заключение

Было создано работоспособное устройство, подстраивающееся индивидуально под каждого пользователя. В комплект к нему было разработано приложение с приятным дизайном и интуитивным управлением, а также возможностью проводить отладку (графическое представление положения датчиков в координатной плоскости). В ходе экспериментов подтвердились основанные на изученных медицинских материалах теории, предложенные для определения правильности осанки.

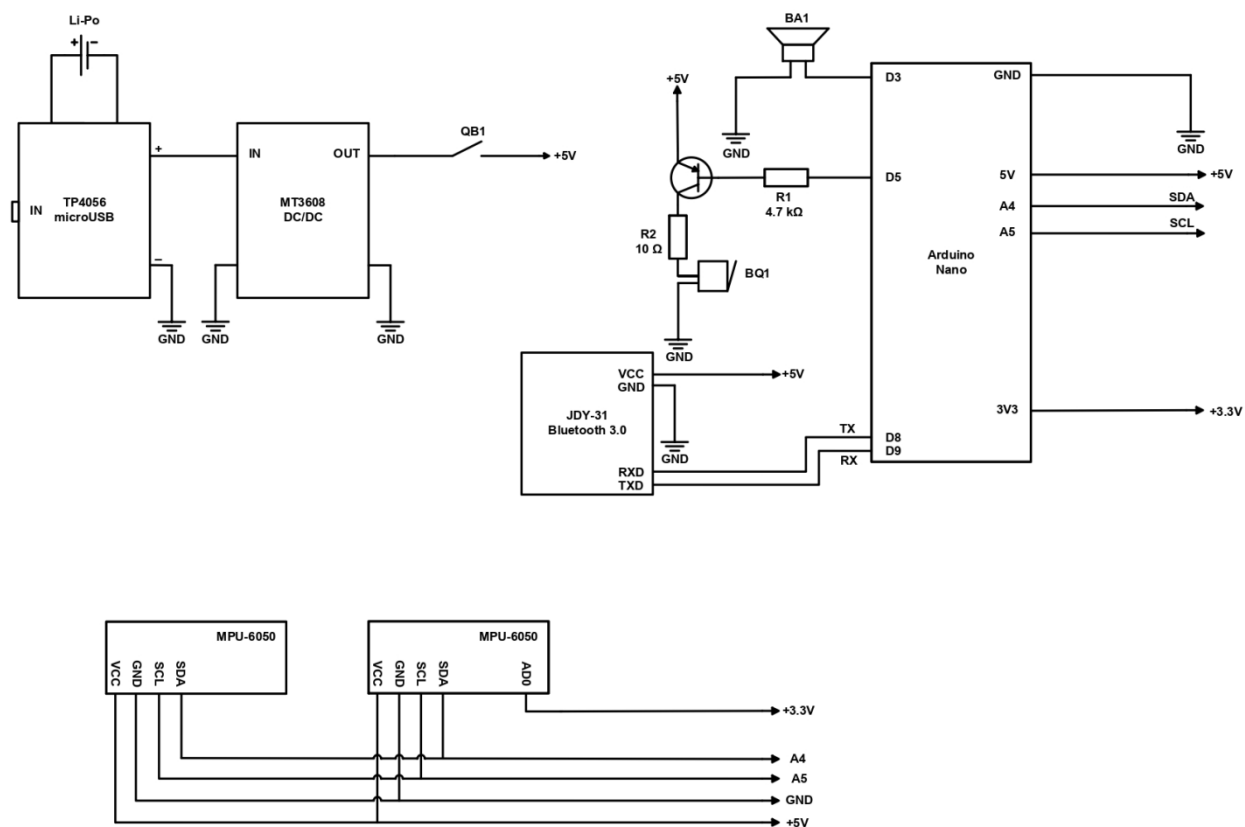
В дальнейшем возможна оптимизация устройства, а также уменьшение его габаритов, путём исключения ненужных элементов (например, в датчике GY-521 есть датчик температуры, который в проекте не используется, а место занимает). Дополнительно планируется реализовать сохранение данных калибровки и настроек в энерго-независимой памяти, которое нужно, чтобы после каждого выключения устройства не требовалось заново проводить настройку.

Список литературы

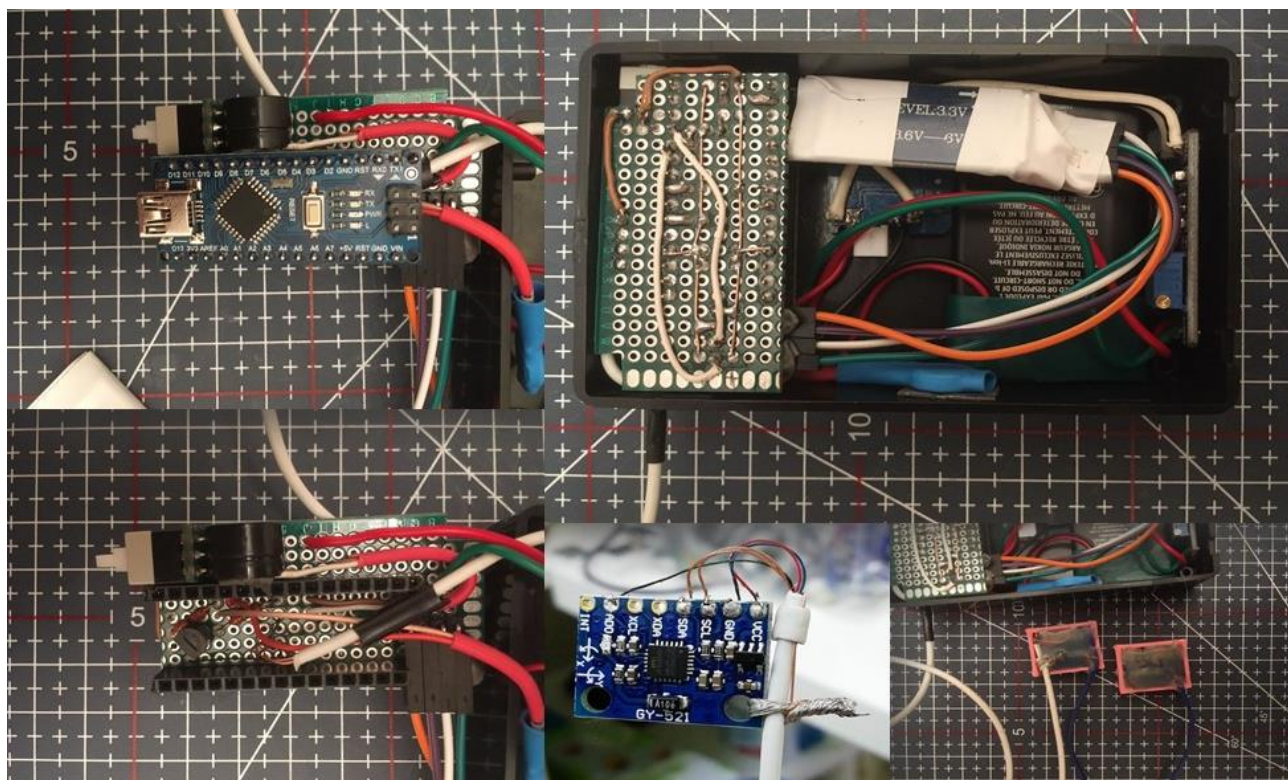
1. Кошелева Л.П. Правильная осанка – залог здоровья человека // Международный журнал прикладных и фундаментальных исследований. – 2014. – № 12-2. – С. 215-217; URL: <https://applied-research.ru/ru/article/view?id=6301> (дата обращения: 01.02.2021).
2. <https://alexgyver.ru/>
3. http://altmed.am/?page_id=4035

Приложения

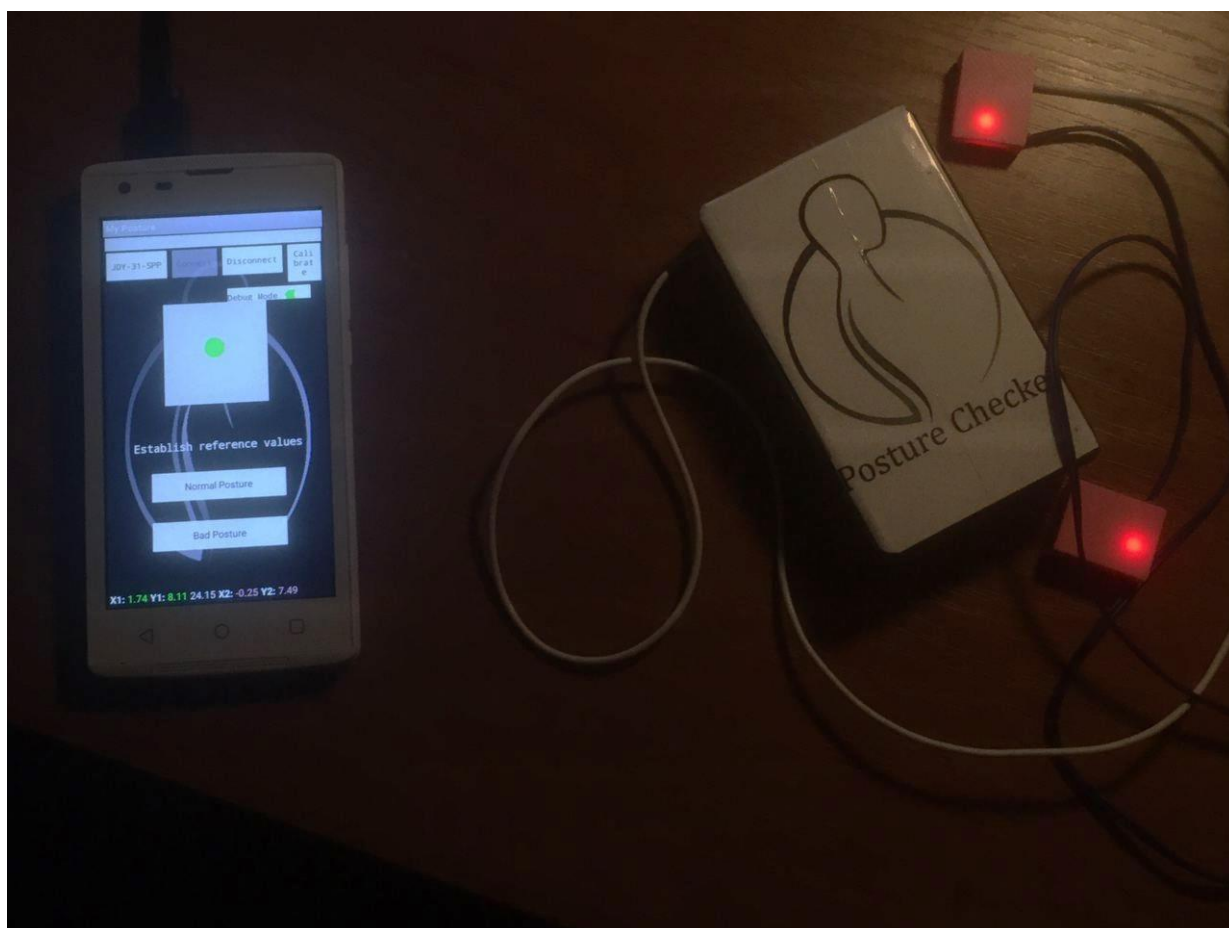
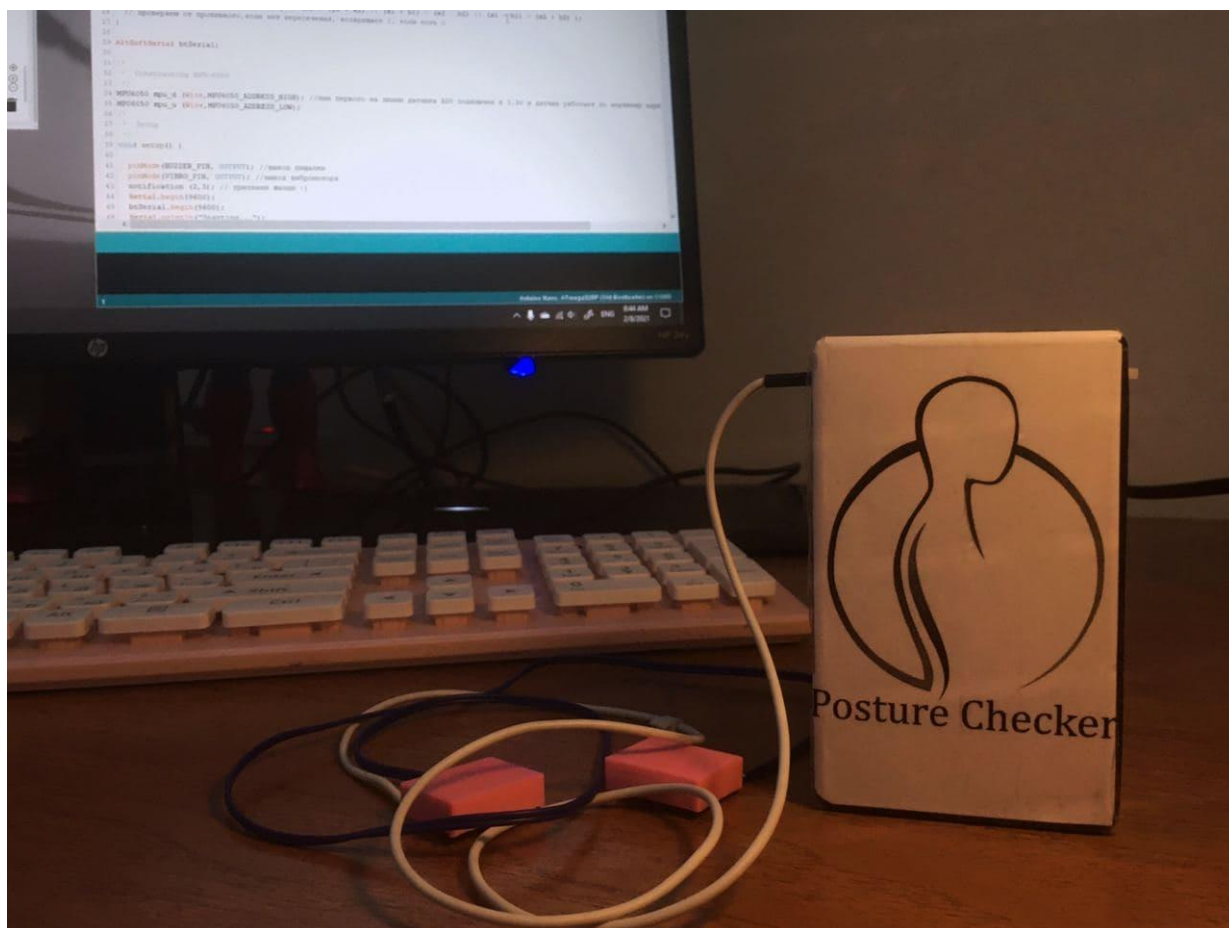
Приложение 1а. Схема устройства электронная принципиальная



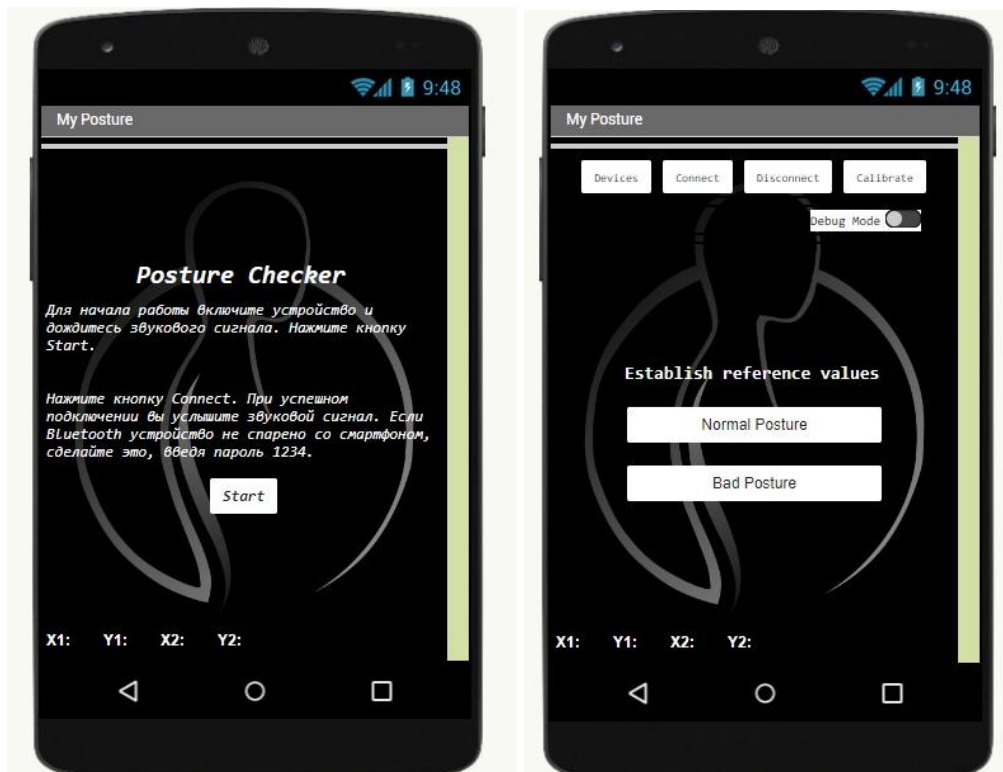
Приложение 1б. Вид устройства изнутри



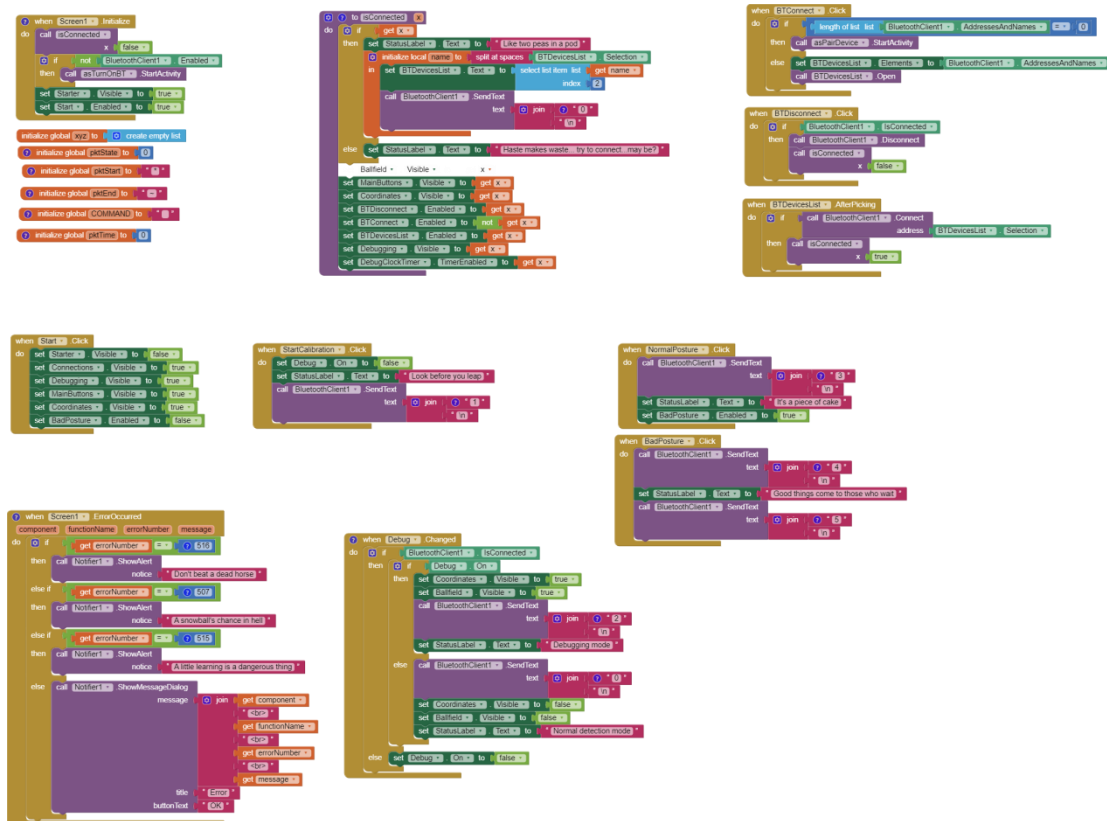
Приложение 2. Внешний вид устройства

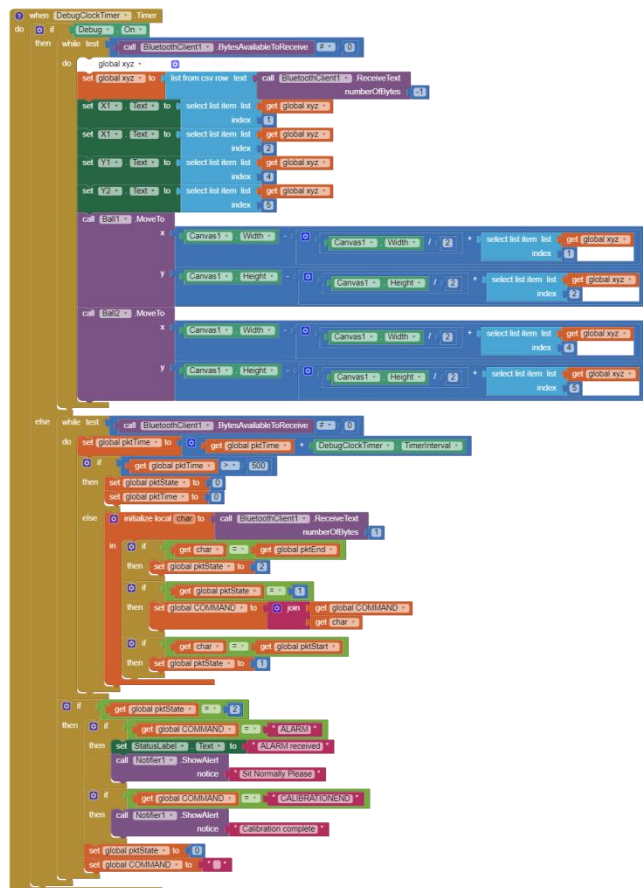


Приложение 3а. Итоговый интерфейс приложения



Приложение 3б. Программа приложения





Приложение 4. Ссылки материалы, созданные и использованные для разработки

1. <https://repl.it/@itsvakonst/ArduinoNanoPostureCheckerCode#main.cpp>
2. <https://repl.it/@itsvakonst/RectangleCheck#main.cpp>
3. <https://drive.google.com/drive/folders/1K97ILNtyJah2m4T6CxMmpbyn7h1mPhv8?usp=sharing>

Приложение 5. Датчики, модель корпуса

