

Государственное учреждение образования  
«Гимназия № 18 г. Минска имени И. С. Миренкова»  
220012, г. Минск, ул. Калинина, 6, (8017)370-74-59

**Практическое применение электронного блока ESP32 с программируемым микроконтроллером в светотехнических устройствах и метода калибровки инерциальных датчиков с применением нейронных сетей и глубокого обучения**

Секция: «Информатика»

**Авторы:**

Комарчук Марк Денисович  
«Гимназия №18 г. Минска  
имени И. С. Миренкова»  
10 «А» класс  
ул. Кнорина 11-14  
тел. +375292095455

Крюк Артемий Данилович  
«Гимназия №18 г. Минска  
имени И. С. Миренкова »  
10 «А» класс  
ул. Восточная, 28-2-43  
тел. +375291851665

**Научный руководитель:**

Леонец Анна Евгеньевна  
«Гимназия №18 г. Минска  
имени И. С. Миренкова»  
учитель физики  
тел. моб.+375291777159

Минск, 2024

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1. ИЗУЧЕНИЕ ПРИМЕНЕНИЯ ПРОГРАММИРУЕМЫХ МИКРОКОНТРОЛЛЕРОВ ДЛЯ СОЗДАНИЯ ЭЛЕКТРОТЕХНИЧЕСКИХ ПРИБОРОВ	5
ГЛАВА 2. КОМПОНЕНТЫ И СХЕМА	6
2.1 ОБРАБОТКА ЗНАЧЕНИЙ ДАТЧИКА УСКОРЕНИЙ (АКСЕЛЕРОМЕТРА) ДЛЯ ПОСЛЕДУЮЩЕГО ОТОБРАЖЕНИЯ НА RGB СВЕТОДИОДНОЙ ЛЕНТЕ	6
2.2 СХЕМА ПИТАНИЯ УСТРОЙСТВА С ВОЗМОЖНОСТЬЮ ЗАРЯДКИ АККУМУЛЯТОРА	7
2.3 ВЫБОР И ИСПОЛЬЗОВАНИЕ RGB СВЕТОДИОДНОЙ ЛЕНТЫ	8
ГЛАВА 3. ПРОГРАММИРОВАНИЕ	9
ГЛАВА 4. ПРОГРАММНАЯ КАЛИБРОВКА ДАТЧИКОВ	11
ЗАКЛЮЧЕНИЕ	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	16
ПРИЛОЖЕНИЕ	17

## **ВВЕДЕНИЕ**

Все, наверное, знают, что ездить на велосипеде по городу очень опасно. Необходимо соблюдать все правила дорожного движения, ведь велосипедист является таким же участником дорожного движения, как автомобилист или мотоциклист. Дети или взрослые, у кого есть городской велосипед, должны быть максимально осторожными. Опасность подстерегает практически везде. Это и автомобили, и пешеходы, и мотоциклы, и так далее.

Несмотря на все проблемы, связанные с ездой на велосипеде, количество велосипедистов в каждом городе неуклонно растет. Это объясняется еще и тем, что на велосипеде вы не будете стоять в пробках, можно просто ее объехать. «Минус» многих больших городов в том, что они не оборудованы велосипедными дорожками. Поэтому, велосипедист участвует в общем движении и должен соблюдать те же правила, что и водители авто, ориентироваться на знаки, обращать внимание на светофоры и пешеходные переходы. Велосипед считается очень удобным видом транспорта. Кроме того, он еще и экологичный, не требует топлива.

### **Актуальность темы**

Езду на велосипеде нельзя назвать безопасной, особенно на оживленных трассах и улицах, где поток машин летает с «бешеной» скоростью, а велосипедисты особенно ничем не защищены, любое столкновение чревато травмой. Единственное, что могут сделать велосипедисты, это привлечь к себе внимание, надев яркую одежду со светоотражающими вставками или обвешиваться лампочками. Для всех любителей поехать на велосипеде ранним утром или поздним вечером, в сумерках предназначен правильный велосипедный шлем.

Так, неоднократно страдая от недоразумений во время движения, мы придумали выход из ситуации. Создали прибор, который будет подавать соответствующие сигналы поворотов, остановок или торможения для других участников дорожного движения. Устройство прикрепляется к обычному шлему, что позволяет водителю, не отрываясь от руля, регулировать транспортные действия, только наклонив голову.

Важным преимуществом «умного» велосипедного шлема является то, что устройство сотрудничает с смартфоном через Bluetooth, что позволяет определить показания и транслировать их сразу же на экране. Интересно, что

здесь можно установить часто используемый маршрут, программа запомнит и поэтому воспроизводить знакомые повороты автоматически. Так, оборудовав свой шлем таким устройством, на улицах города можно будет чувствовать себя уверенно и комфортно даже на велосипеде.

**Цель:** создание устройства изменяющего цвет RGB светодиодной ленты в зависимости от ускорения с использованием микроконтроллера и датчика ускорений, изучение применения технологии нейронных сетей для калибровки инерциальных датчиков

**Задачи:**

1. изучить применение программируемых микроконтроллеров для создания электротехнических приборов;
2. рассмотреть возможность обработки значений датчика ускорений (акселерометра) и гироскопа для последующего отображения на RGB светодиодной ленте эффекта торможения;
3. рассмотреть возможность программного управления цветовым отображением на RGB светодиодной ленте;
4. компоновка элементов устройства и оценка эффективности работы светодиодной ленты при изменении скорости передвижения.

**Объект исследования:** возможность применения микроконтроллеров для создания устройств, повышающих безопасность передвижения на дорогах.

**Предмет исследования:** электронный блок ESP32 с программируемым микроконтроллером.

**Методы исследования:** изучение теоретической литературы, анализ и обобщение изученного материала, использование языка C++ для программирования устройств на микроконтроллерах, изучение и анализ показаний акселерометра и гироскопа во время торможения и поворота, проведение экспериментов для оценки работы светодиодной ленты, компоновка и монтаж элементов оборудования.

# ГЛАВА 1. ИЗУЧЕНИЕ ПРИМЕНЕНИЯ ПРОГРАММИРУЕМЫХ МИКРОКОНТРОЛЛЕРОВ ДЛЯ СОЗДАНИЯ ЭЛЕКТРОТЕХНИЧЕСКИХ ПРИБОРОВ

**Микроконтроллер** – это специальная микросхема, предназначенная для управления различными электронными устройствами. Разработчики микроконтроллеров придумали остроумную идею – объединить процессор, память, ПЗУ и периферию внутри одного корпуса, внешне похожего на обычную микросхему.

Оптимальным вариантом в отношении цена/качество/доступность была Макетная плата ESP32 - небольшая плата с собственным процессором и памятью. Большую популярность плата ESP32 приобрела не только из-за низкой стоимости, легкости разработки и программирования, но, главным образом, благодаря наличию плат расширения (так называемых шилдов), добавляющих ESP32 дополнительную функциональность. Шилды подключаются к ESP32 с помощью имеющихся на них штыревых разъемов. Существует много версий платы ESP32. Для реализации моего проекта была выбрана плата ESP32 wroom (рисунок 1).

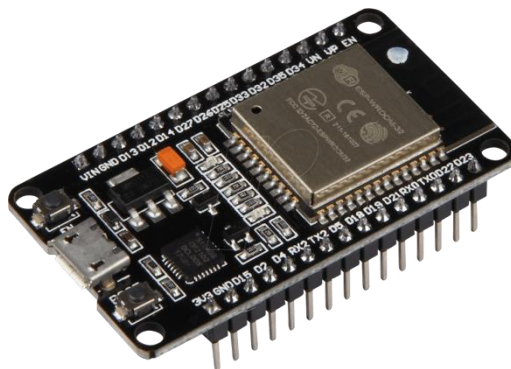


Рисунок 1 ESP32 WROOM

## Технические характеристики платы ESP32 WROOM:

### Микроконтроллер и управление:

Tensilica Xtensa LX6 двухъядерный (или одноядерный) 32-разрядный процессор, с тактовой частотой 160 или 240 МГц и производительностью до 600 DMIPS

### Память:

- SRAM 520KB
- Кибит Флэш-память 448KB

- NVRAM 16KB

#### **Беспроводная связь:**

- Wi-Fi
- Bluetooth

#### **Периферийные интерфейсы:**

- 12-разрядный АЦП до 18 каналов
- 2×8 бит ЦАП
- 10×портов для подключения
- 4×SPI мастер-интерфейса
- 2×I2S мастер-интерфейса
- 2×I2C мастер-интерфейса
- 3×UART интерфейса
- Ethernet MAC interface

#### **Управление питанием:**

- Рабочее напряжение от 2,2—3,6 В
- Максимальная скорость передачи данных 150 Мбит/с при 11n HT40, 72 Мбит/с при 11n HT20, 54 Мбит/с @ 11g, и 11 Мбит/с при 11b
- Устойчивая пропускная способность UDP 135 Мбит/

## **ГЛАВА 2. КОМПОНЕНТЫ И СХЕМА**

### **2.1 ОБРАБОТКА ЗНАЧЕНИЙ ДАТЧИКА УСКОРЕНИЙ (АКСЕЛЕРОМЕТРА) ДЛЯ ПОСЛЕДУЮЩЕГО ОТОБРАЖЕНИЯ НА RGB СВЕТОДИОДНОЙ ЛЕНТЕ**

Определившись с “мозгом”, мы перешли к выбору датчиков, элемента питания, индикации шлема, а также разработке схемы соединения всех элементов.

**Акселерометр** (лат. *Accelero* — ускоряю и др.-греч. *μετρέω* «измеряю») — прибор, измеряющий проекцию кажущегося ускорения (разности между истинным ускорением объекта и гравитационным ускорением). Как правило, акселерометр представляет собой чувствительную массу, закреплённую в упругом подвесе. Отклонение массы от её первоначального положения при наличии кажущегося ускорения несёт информацию о величине этого ускорения. По конструктивному исполнению акселерометры подразделяются на

однокомпонентные, двухкомпонентные, трёхкомпонентные. Соответственно, они позволяют измерять ускорение вдоль одной, двух и трёх осей.

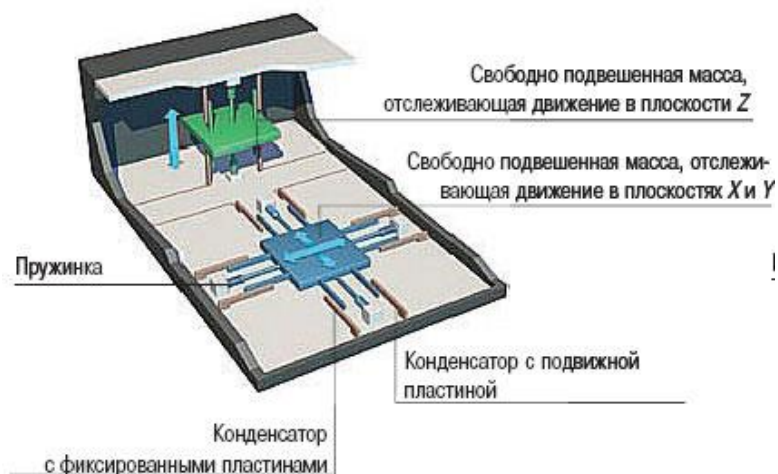


Рисунок 2

В качестве датчика был выбран модуль GY-521 это довольно распространенная микросхема имеющая сразу и гироскоп и акселерометр, для связи с модулем используется шина I2C (Inter-Integrated Circuit), позволяющая подключать до 119 устройств и использующая восьмибитную адресацию (127 адресов, из которых 8 служебных) для выбора конкретного устройства. В программе адрес представлен в виде HEX числа (пример: 0x27, 0x77).



Рисунок 3. Модуль Gy 521

## 2.2 СХЕМА ПИТАНИЯ УСТРОЙСТВА С ВОЗМОЖНОСТЬЮ ЗАРЯДКИ АККУМУЛЯТОРА

Прибор измеряет и сообщает микроконтроллеру данные об ускорении и ориентации.

Для питания всей схемы было принято решение использовать аккумулятор формата 18650, аккумуляторы этого форм фактора на выходе отдают напряжение равное 3.7V при максимальном заряде, однако если оставить просто аккумулятор, он может попасть в глубокий разряд, то есть

напряжение упадет ниже 2V. Для предотвращения такой ситуации, а также для зарядки аккумулятора было принято решение установить зарядную плату.

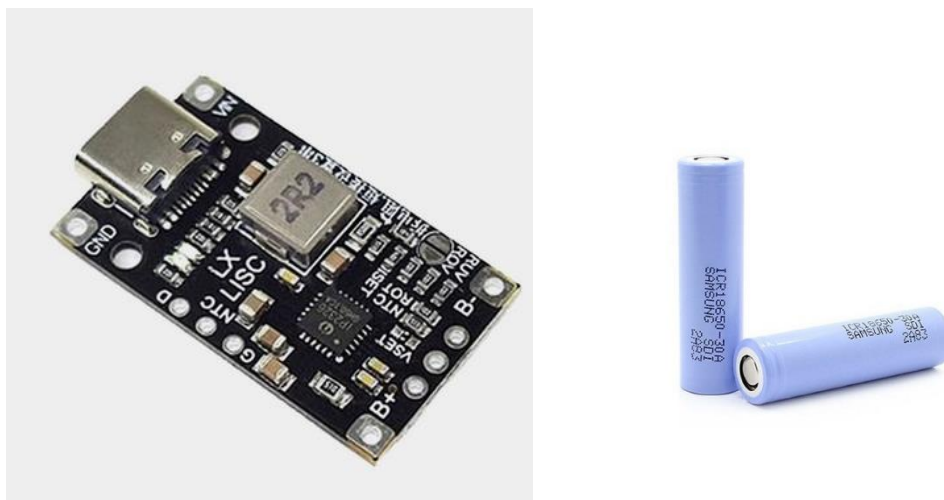


Рисунок 4 Зарядная плата и аккумулятор 18650

Также необходимо добавить переключатель чтобы иметь возможность выключать устройство не доставая аккумулятор.

## 2.3 ВЫБОР И ИСПОЛЬЗОВАНИЕ RGB СВЕТОДИОДНОЙ ЛЕНТЫ

**RGB светодиодная лента.** На этой ленте стоят RGB светодиоды. Такой светодиод имеет уже 4 выхода, один общий +12 (анод), и три минуса (катода) на каждый цвет, т.е. внутри одного светодиода находится три светодиода разных цветов. Соответственно такие же выходы имеет и лента: 12, G, R, B. Подавая питание на общий 12 и любой из цветов, мы включаем этот цвет. Подадим на все три – получим белый, зелёный и красный дадут жёлтый, и так далее(Рисунок 5).

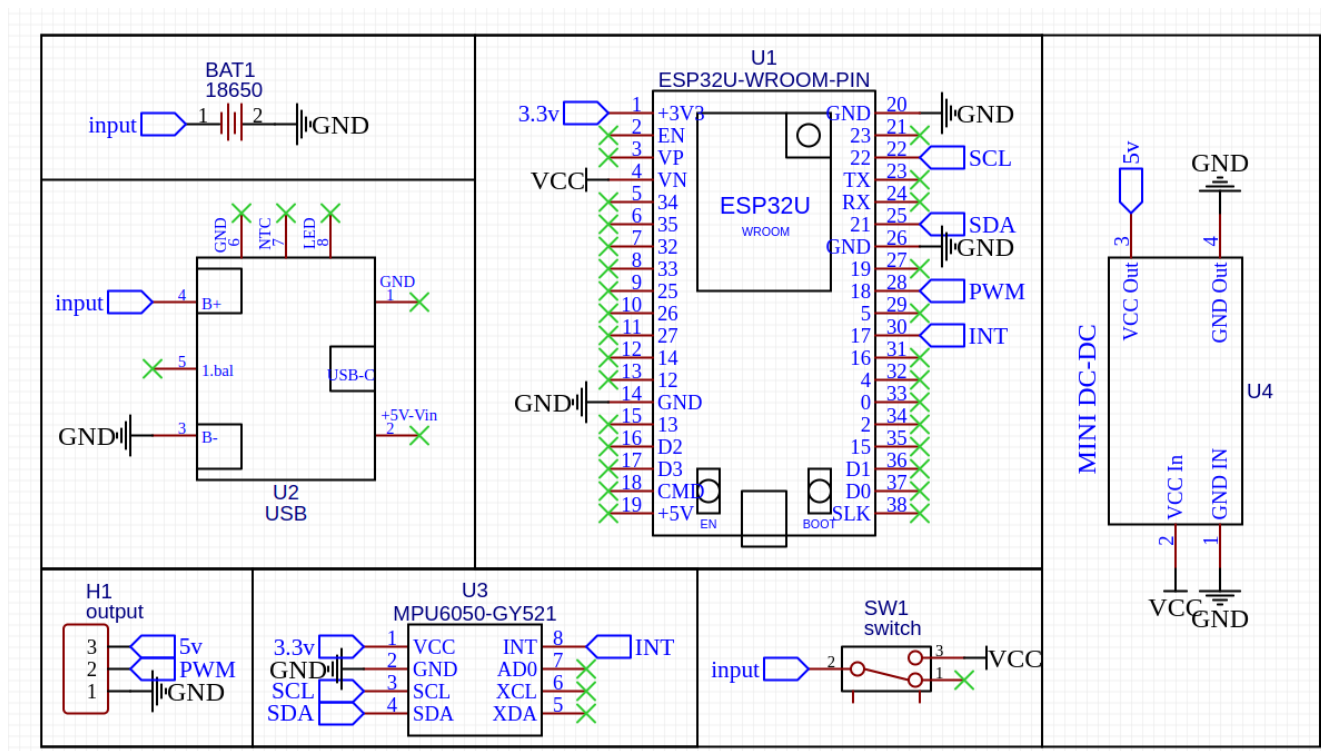


Рисунок 5

**Адресная светодиодная лента,** вершина эволюции лент. Представляет собой ленту из адресных диодов, один такой светодиод состоит из RGB



светодиода и контроллера. Да, внутри светодиода уже находится контроллер с тремя транзисторными выходами. Благодаря такой начинке у нас есть возможность управлять цветом (то бишь яркостью r g b) любого светодиода в ленте и создавать потрясающие эффекты. Адресная лента может иметь 3-4 контакта для подключения, два из них всегда питание (5V и GND), и остальные (один или два) – логические, для управления.



Для наглядности схемы, а также удобства при сборке было составлена схема с помощью EasyEDA.

Так же в будущем, если проект получит развитие, можно сделать печатные платы, которые необходимы для упрощения пайки деталей, а также придания товарного вида устройству.

### ГЛАВА 3. ПРОГРАММИРОВАНИЕ

Использование каждого элемента системы, потребовало тщательного исследования и чтения огромного количества литературы. Однако и это еще не все, даже если правильно подключить и скомпоновать элементы устройства, оно не будет работать без программной части: контроллер для начала нужно запрограммировать.

Алгоритм по которому будет осуществляться управление разделим на две части: на стороне устройства(ESP32) и на стороне телефона(для более точного определения поворота и торможения с помощью нейросети) — связь реализуем через WI-FI, который поддерживается как ESP, так и любым мобильным телефоном. Мы выбрали WI-FI, а не Bluetuthe, по причине большей скорости передачи данных, что важно для нашей задачи. Также разработаем приложение под Android для вывода некоторой информации и в том числе для отладки.

Низкоуровневая часть: под низкоуровневой частью подразумеваю программирование ведомого устройства на C++. Для прошивки платы и написания программы будем использовать расширение PlatformIO для популярной программы Visual Studio Code. Можно было использовать и Arduino IDE, но в связи большим опытом в PlatformIO.

- 1) реализуем работу с Gy521 через библиотеку <MPU6050.h>

Инициализация:

```
void setup() {
  Wire.begin();
  Serial.begin(9600);
  //mpu
  mpu.initialize();
  // состояние соединения
  Serial.println(mpu.testConnection() ? "MPU6050 OK" : "MPU6050 FAIL");
}
```

Получение данных:

```
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
mp[0] = ax/160;
mp[1] = ay/160;
mp[2] = az/160;
mp[3] = gx;
mp[4] = gy;
mp[5] = gz;
delay(5);
Serial.print(mp[0]); Serial.print('\t');
Serial.print(mp[1]); Serial.print('\t');
Serial.print(mp[2]); Serial.print('\t');
Serial.print(mp[3]); Serial.print('\t');
Serial.print(mp[4]); Serial.print('\t');
Serial.println(mp[5]);
```

- 2) работа с адресной лентой через библиотеку <FastLED.h>:

```

110 void stop(bool flag){
111     byte j =0;
112     for(int i = 0; i < LED_NUM1; i++){
113         leds1[i].setRGB(255, 255, 255);
114     }
115     if(flag){
116         for(int i = 0; i < LED_NUM1; i++){
117             leds1[i].setRGB(0, 255, 0);
118         }
119     }
120 }

```

## ГЛАВА 4. ПРОГРАММНАЯ КАЛИБРОВКА ДАТЧИКОВ

**Проблематика:** фактические показания электронных датчиков могут не отвечать физическим законам движения объекта (велосипедиста). Вследствие чувствительности прибора, возникает так называемый «эффект шума датчиков» и, как результат, некорректная работа индикации шлема. Для решения данной проблемы могут быть применены программные методы фильтрации данных. Одним из них является метод совместной калибровки инерциальных датчиков микроконтроллера с применением нейронных сетей.

**Цель:** разработать эффективную систему калибровки инерциальных датчиков на основе технологии нейронных сетей с применением алгоритмов машинного обучения.

**Преимуществами** данной технологии по сравнению с классическим методом комплементарного фильтра является повышенная точность выходных данных за счёт автоматического подбора коэффициентов взвешенной суммы, осуществляемого при помощи технологии обучения нейронной сети.

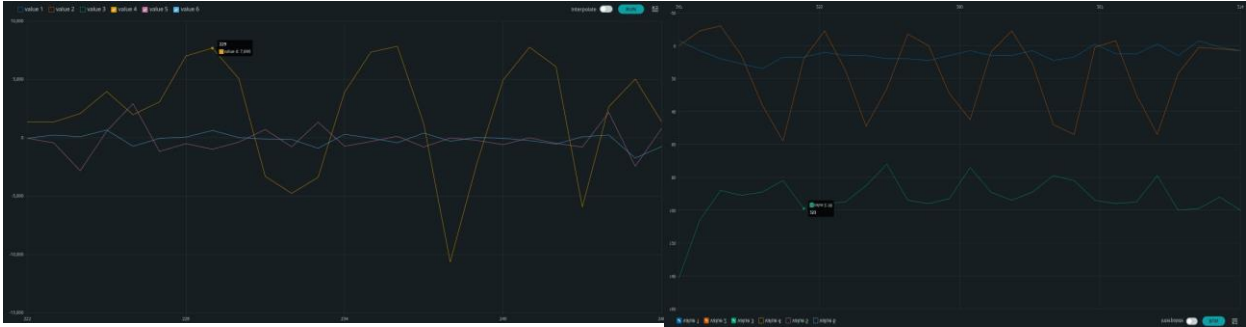
Для начала, ознакомимся с принципами расчёта фактических значений углов наклона объекта относительно неподвижной оси на основе данных акселерометра и гироскопа.

Как известно, акселерометр возвращает значения изменения ускорения по осям (X, Y, Z). Обозначим его показания как  $a_x$ ,  $a_y$  и  $a_z$  соответственно. Задачей гироскопа является определение угловой скорости по трём заданным осям. Примем получаемые данные за  $G_x$ ,  $G_y$ ,  $G_z$ . Руководствуясь справочными данными, найдём необходимое значение угла по показаниям каждого из приборов:

$$\alpha_{acc} = \arctg \left( \frac{a_y}{\sqrt{a_x^2 + a_z^2}} \right)$$

$$\alpha_{sup} = \alpha_0 + G_y$$

Проведя серию экспериментов и обработав получаемые данные согласно (1) и (2), обнаружим серьёзные расхождения значений угла в определениях каждым прибором по отдельности.



Решение данной проблемы кроется в создании эффективного цифрового фильтра входных данных (т. е. алгоритма, с помощью которого будет удобно автоматически корректировать получаемое значение искомого угла).

Существует множество методов аппроксимации получаемых результатов, и в ходе реализации проекта было решено использовать алгоритм машинного обучения нейронной сети (основанный на элементах метода взвешенной суммы). Для создания необходимого алгоритма в виде нейронной сети будем использовать так называемую «стандартную схему» архитектуры программы, основанную на технологии сигмовидных нейронов.

Так как алгоритм подразумевает автоподбор весов и смещений для корректного определения угла, в процессе создания необходимо выбрать метод обучения нейронной сети. Мы используем метод градиентного спуска с применением обратного распространения ошибки. Входными данными для обучения будем считать экспериментальные показания акселерометра и гироскопа, а также вычисленный по комплементарному методу на основе данных значений прогнозируемый угол.

Используя стандарт языка программирования Python 3.12, запишем алгоритмическое представление нейронной сети в интерпретаторе PyPy 7.3.17:

```

def feedforward(self, a):
    for b, w in zip(self.biases, self.weights):
        a = sigmoid(np.dot(w, a)+b)
    return a

def SGD(self, training_data, epochs, mini_batch_size, eta,
        test_data=None):
    if test_data: n_test = len(test_data)
    n = len(training_data)
    for j in xrange(epochs):
        random.shuffle(training_data)
        mini_batches = [
            training_data[k:k+mini_batch_size]
            for k in xrange(0, n, mini_batch_size)]
        for mini_batch in mini_batches:
            self.update_mini_batch(mini_batch, eta)
        if test_data:
            print "Epoch {0}: {1} / {2}".format(
                j, self.evaluate(test_data), n_test)
        else:
            print "Epoch {0} complete".format(j)

def update_mini_batch(self, mini_batch, eta):
    nabla_b = [np.zeros(b.shape) for b in self.biases]
    nabla_w = [np.zeros(w.shape) for w in self.weights]
    for x, y in mini_batch:
        delta_nabla_b, delta_nabla_w = self.backprop(x, y)
        nabla_b = [nb+dnb for nb, dnb in zip(nabla_b, delta_nabla_b)]
        nabla_w = [nw+dnw for nw, dnw in zip(nabla_w, delta_nabla_w)]
    self.weights = [w-(eta/len(mini_batch))*nw
                     for w, nw in zip(self.weights, nabla_w)]
    self.biases = [b-(eta/len(mini_batch))*nb
                    for b, nb in zip(self.biases, nabla_b)]

def backprop(self, x, y):
    nabla_b = [np.zeros(b.shape) for b in self.biases]
    nabla_w = [np.zeros(w.shape) for w in self.weights]
    # feedforward
    activation = x
    activations = [x] # list to store all the activations, layer by layer
    zs = [] # list to store all the z vectors, layer by layer
    for b, w in zip(self.biases, self.weights):
        z = np.dot(w, activation)+b
        zs.append(z)
        activation = sigmoid(z)
    return activations, zs, nabla_b, nabla_w

```

Рисунок 1. Пример интерпретации нейронной сети в виде кода

Проведем некоторое количество экспериментов с инерциальными датчиками в лабораторных условиях и полученные данные оформим в виде таблицы:

Таблица 1. База входных данных для обучения нейронной сети

$a_x$	$a_y$	$a_z$	$G_x$	$G_y$	$G_z$	$\alpha_{\text{датч}}$
10	46	86	2775	-361	-172	-357.11
3	38	84	1099	418	45	414.06
11	29	90	1164	246	457	243.71
14	36	86	499	203	320	201.19
1	21	95	1827	57	-429	56.55
1	19	87	1021	227	961	224.85
3	-3	95	1396	-41	-30	-40.60

Как было уже упомянуто, вследствие «эффекта шума» датчиков, прогнозируемое значение угла поворота, высчитанное по формулам (1) и (2) соответственно для акселерометра и гироскопа, может существенно расходиться с фактическим значением угла наклона.

Обучив описанный в работе нейросетевой алгоритм на основе данных Таблицы 1, применяем его для корректировки значения угла поворота:

$a_x$	$a_y$	$a_z$	$G_x$	$G_y$	$G_z$	$\alpha_{\text{нейрон}}$
10	46	86	2775	-361	-172	67.83
3	38	84	1099	418	45	55.66
11	29	90	1164	246	457	43.64
14	36	86	499	203	320	51.43
1	21	95	1827	57	-429	36.33
1	19	87	1021	227	961	31.29
3	-3	95	1396	-41	-30	64.5

После завершения цифровой обработки данных во всех  $n=7$  экспериментах итоговое значение угла наклона находится в допустимом диапазоне, что свидетельствует о корректной работе алгоритма.

Вследствие особенностей машинного обучения, алгоритм цифрового фильтра автоматически настраивает веса и смещения в функции, что доказывает эффективность данного способа обработки.

Основываясь на этих данных, можем заявить об успешном достижении цели данной части работы.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения работы:

- разработано и собрано устройство, позволяющее стать заметнее на дороге;
- исследованы показания акселерометра и гироскопа при торможении и повороте;
- рассмотрена возможность программного управления цветовым отображением на RGB светодиодной ленте в зависимости от показаний датчика (акселерометра);

В результате данной работы разработан шлем, который обеспечит более безопасное передвижение на велосипеде.

Преимущества данного устройства:

- шлем не просто защищает голову, но и делает вас более заметными, благодаря встроенным светодиодам, которые автоматически включаются при недостаточной освещенности и меняют цвет при торможении и указывают на поворот;
- компактность: вес около 150 г, размер 5x10 см;
- данное устройство не вызывает дискомфорта и не мешает управлению при его использовании.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Модуль GY-521// Паяльник [Электронный ресурс]. – 2010. Режим доступа: <http://cxem.net/mc/mc324.php/>
2. RGB лента со встроенным ШИМ// Портал светозффектов [Электронный ресурс]. – 2010. Режим доступа: [http://lightportal.at.ua/publ/rgb/mnogocvetnaja\\_svetodiodnaja\\_podsvetka\\_ili\\_rgb\\_kontroller/17-1-0-53/](http://lightportal.at.ua/publ/rgb/mnogocvetnaja_svetodiodnaja_podsvetka_ili_rgb_kontroller/17-1-0-53/)
3. Michael A. Nielsen, " Neural Networks and Deep Learning", Determination Press, 2015
4. Плата ESP32 WROOM// Портал плат [Электронный ресурс]. -2020. Режим доступа: <https://wiki.amperka.ru/products:esp32-wroom-wifi-devkit-v1>

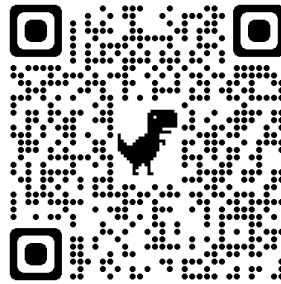


## ПРИЛОЖЕНИЕ 1

Для ознакомления с проектом на github перейдите по ссылке или отсканируйте QR-код.

Ссылка: <https://github.com/SaShiro228/helmet>

QR-код:



### Интеллектуальный велошлем

