

Проект GloK

Выполнил: Булыгин Н.С.

# Оглавление

<b>1</b>	<b>Введение</b>	<b>2</b>
1.1	Цель проекта . . . . .	2
1.2	Рассмотренные технологии для реализации . . . . .	2
<b>2</b>	<b>Финансовые затраты</b>	<b>4</b>
<b>3</b>	<b>Архитектура</b>	<b>5</b>
3.1	основной принцип работы . . . . .	5
3.2	Электрическая схема *При соединении микроконтроллера с каждым датчиком . . . . .	7
3.3	Программная часть . . . . .	7
3.3.1	Данные с датчиков . . . . .	7
3.3.2	Кадры . . . . .	7
3.3.3	Вычисление перемещения . . . . .	7
3.4	Преобразование в команды . . . . .	7
3.4.1	Преобразования в нажатия клавиш . . . . .	7
3.4.2	Преобразования в геймпад . . . . .	11
3.4.3	Преобразования в компьютерную мышь . . . . .	11
3.5	Конфигурация . . . . .	11
3.5.1	Заводские настройки . . . . .	12
3.5.2	Настройки пользователя . . . . .	12
3.6	Bluetooth . . . . .	13
3.6.1	API . . . . .	13
3.6.2	Справка по командам . . . . .	13
3.6.3	Конечный вывод команд . . . . .	13

# Глава 1

## Введение

### 1.1 Цель проекта

В качестве цели проект перед собой ставит улучшение взаимодействия пользователя с цифровой техникой. На данный момент в качестве основных средств взаимодействия с цифровой техникой используются периферийные устройства такие как: клавиатура, компьютерная мышь, различного рода и различного рода геймпады. Предлагается создать конфигурируемое периферийное устройство которое позволит объединить различного рода периферийные устройства и взаимодействовать с широким рядом цифровой техники

### 1.2 Рассмотренные технологии для реализации

При проектировании устройства были рассмотрены следующие технологии: нейронная связь, экзоскелет, датчики гироскопа-акселерометра

**Нейронные датчики** один из лучших вариантов для реализации данного проекта т.к. они могут предоставить широкий диапазон входных данных сочетающиеся со скоростью работы. Данный вариант было принято **не использовать по причине дороговизны** конечного продукта

В качестве альтернативы нейронным датчикам рассматривались **датчики мышечной активности**, однако от них было принято решение так же отказаться по причине того, что данный способ не мог обеспечить достаточный диапазон данных для последующей обработки Система посредством экзоскелета имеет огромное количество преимуществ в сравнении с другими способами реализации данного продукта. Рассматриваемая система может предоставить большую вариативность данных. Однако глав-

ным плюсом может быть внедрение обратной связи в систему. В частности, под обратной связью подразумевается способность системы давать оператору сигналы о происходящем. Например, с помощью электродвигателей создавать сопротивление для перемещения позиции. Отказаться от данной системы было принято по **причинам больших габаритов будущей конструкции и усложнения реализации проекта**.

Наконец, было принято решение для реализации системы на основе **датчиком гироскопа-акселерометра**. Данное решение сочетало в себе компактность будущего изделия и его дишивизна с большим количеством возможных выходных данных.

## Глава 2

# Финансовые затраты

## Глава 3

# Архитектура

### 3.1 основной принцип работы

Конечный продукт предполагается для размещения на руках - все датчики на пальцах и главные платы на выше запястья, на лучевой кости. Всего было принято решение разместить 10 датчиков на руке, что также отображено на рис 3.1:

1. на тыльной стороне ладони (далее датчик №0)
2. На Проксимальной фаланге пальцев:
  - (a) Большого пальца (далее датчик №1)
  - (b) Указательного пальца (далее датчик №2)
  - (c) Среднего пальца (далее датчик №3)
  - (d) Безымянного пальца (далее датчик №4)
  - (e) Мизинца (далее датчик №5)
3. На Дистальных фалангах
  - (a) Указательного пальца (далее датчик №6)
  - (b) Среднего пальца (далее датчик №7)
  - (c) Безымянного пальца (далее датчик №8)
  - (d) Мизинца (далее датчик №9)

Рис. 3.1: Расположение датчиков на руке (Справа - прототип)

Для прототипа в качестве считывающих датчиков было решение использовать MPU6050 - 3-х осевой гироскоп и 3-х же осевой акселерометр в одном корпусе. На момент разработки проекта не было известно какой алгоритм будет использоваться для преобразования данных в команды поэтому прототип с запасом мощностей был построен на базе платы ESP32-WROOM-32-DevKit-V1.

Питание микроконтроллера и всех датчиков будет от батарейки-кроны 9V. Не предполагается использовать встроенные аккумуляторы по следующим причинам:

1. Неудобство использования. Когда заряд аккумулятора подходит к концу, пользователю необходимо прекратить использование устройства. При питании от батареек, пользователю достаточно заменить их и продолжить использование
2. Долговечность устройства. Аккумуляторы при благоприятных условиях в среднем служат до 8ми лет. В будущем замена источника питания будет осложнена установкой нового аккумулятора

## 3.2 Электрическая схема \*При соединении микроконтроллера с каждым датчиком

## 3.3 Программная часть

### 3.3.1 Данные с датчиков

Трех осевой гироскоп и акселерометр MPU6050 позволяет нам получать 6 данных: угол поворота и ускорение по трем осям. Данные необходимо подвергать различным трансформациям (например преобразование Фурье) для снижения влияния шумов

### 3.3.2 Кадры

Данные с датчиков необходимо разбивать на отдельные фрагменты которые далее будут использованы. Для кадра предлагается использовать систему схожую с векторами. Например преобразовать данные к виду: При углах X,Y,Z было перемещение на X,Y,Z. При этом под смена вектора будет происходить, когда ускорение по одной из осей будет более чем какое-то пороговое значение для срабатывания. Из очевидных недостатков можно выделить, что система не будет реагировать на медленные движение, соответственно у которых ускорение будет меньше чем пороговое значение. Ожидаемые выходные данные: [УголX, УголY, УголZ,ПеремещениеX, ПеремещениеY, ПеремещениеZ]

### 3.3.3 Вычисление перемещения

## 3.4 Преобразование в команды

При разработке были рассмотрены алгоритмический подход и подход проектирования искусственного интеллекта. От последнего было принято решение отказаться, так как при добавлении каждой новой команды необходимо будет переучивать систему, что негативно скажется на возможностях для расширения.

### 3.4.1 Преобразования в нажатия клавиш

В систему заранее должны быть определены жесты которые соответствуют определенной команде. Далее последовательность данных предлагается преобразовывать в команды по следующему алгоритму:



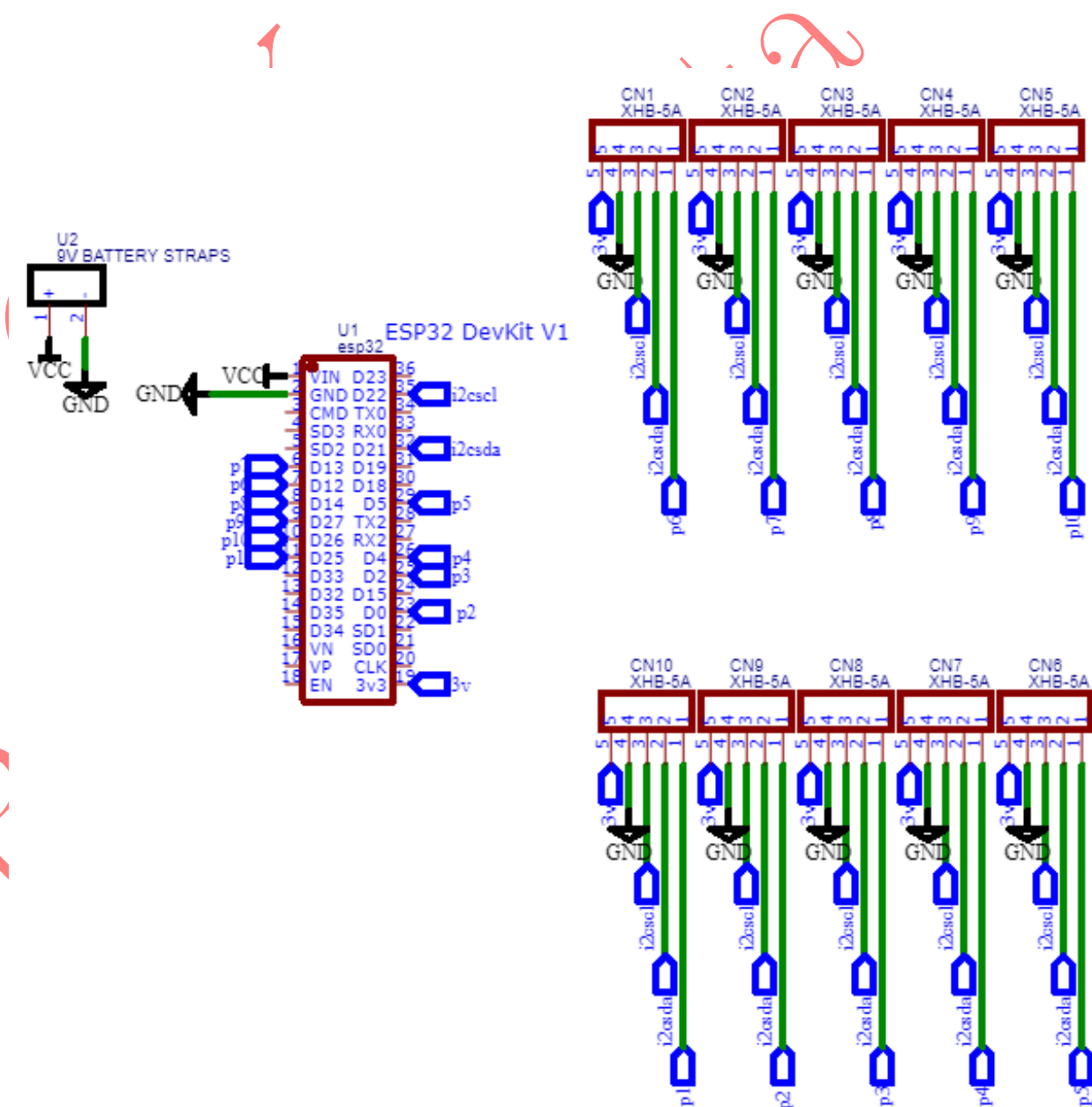


Рис. 3.2: Принципиальная схема подключения платы микроконтроллера

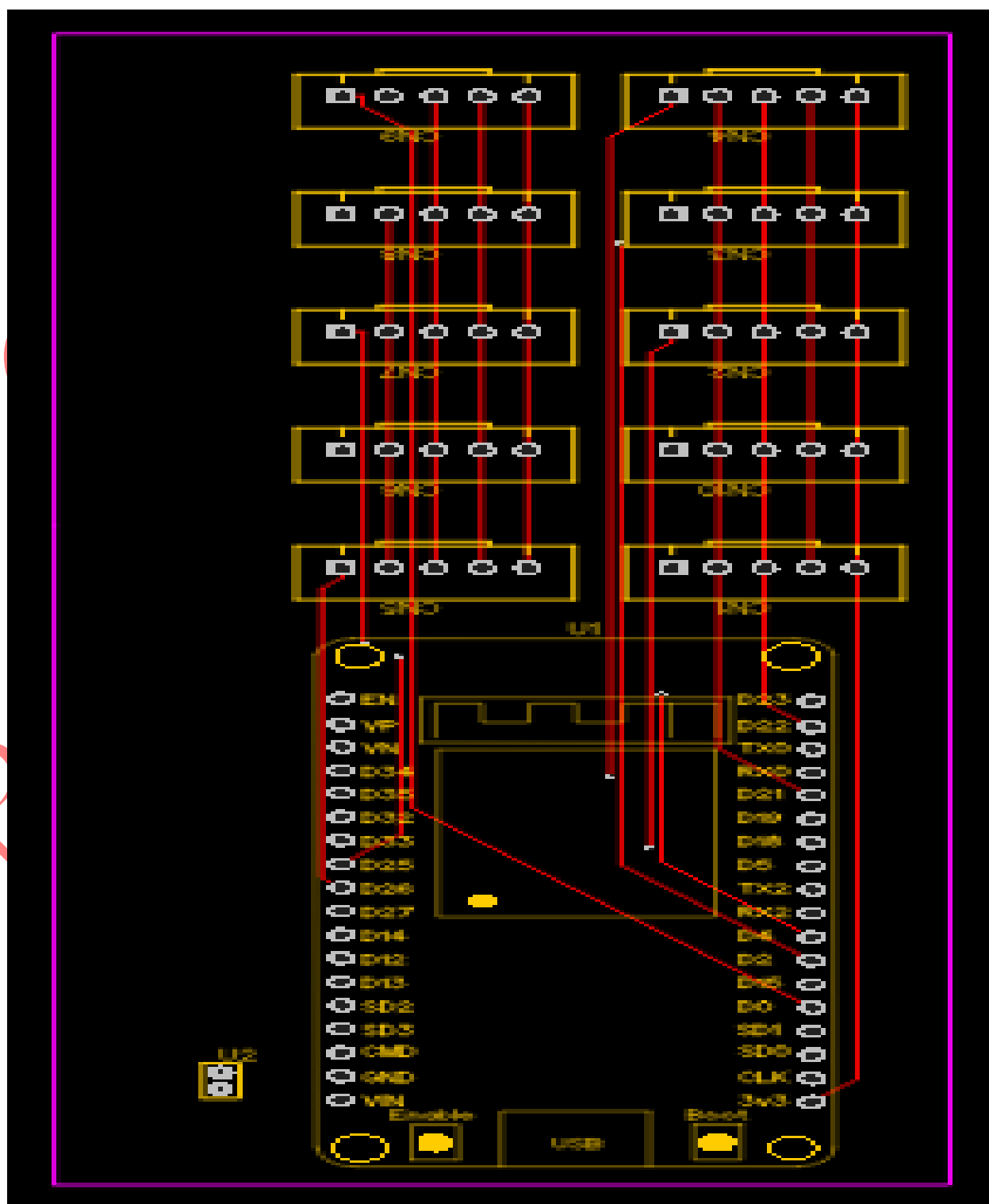


Рис. 3.3: Электрическая схема подключения платы микроконтроллера

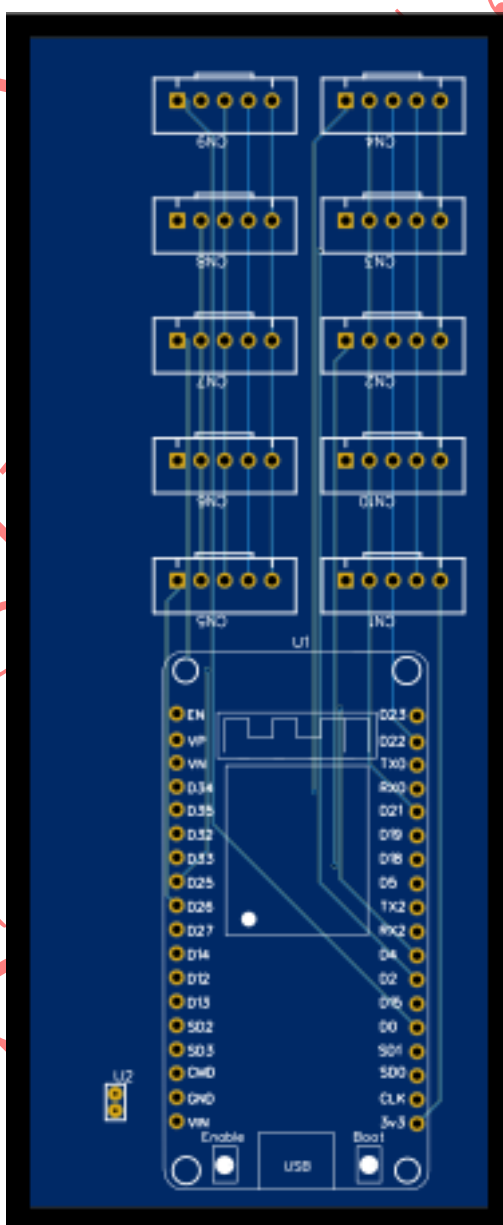


Рис. 3.4: Изображение платы микроконтроллера

1. Создается скользящее окно размерностью максимальной длины жестов( $N$ ) и указатель на начальную позицию в последовательности( $S$ )
2. Создается цикл с 0 по  $N$  с итератором  $i$ :
  - (a) Считываются данные с  $S$  по  $N-i$  и передаются на проверку корреляции с жестами которые имеют  $N-i$  данных в последовательности:
    - i. Если значение корреляции превышает некоторое пороговое значение, то выдать команду о нажатии соответствующей клавиши. Передвинуть указатель на позицию  $S + N-i$  и завершить цикл
  - (b) Если не найдено совпадений то передвинуть указатель на позицию  $S+N$
3. Клавишу считать нажатой до тех пор, пока не встретится очередной вектор

### 3.4.2 Преобразования в геймпад

Для преобразования в геймпад выбирается определенный датчик и задается заранее необходимы диапазон значений. Далее происходит преобразование углов датчика в заданные значения по каждой из осей

### 3.4.3 Преобразования в компьютерную мышь

Выбирается определенный датчик согласно которому будет происходить считывание изменений. Предлагается два способа обработки данных:

1. Курсор мыши будет смещаться согласно разнице между начальным и текущим углом определенного датчика
2. Курсор мыши будет жестко привязан к образованию геймпада где крайние положения соответствуют краям экрана

## 3.5 Конфигурация

В устройстве предусмотрена возможность конфигурации устройства под различные форматы конечных команд

### 3.5.1 Заводские настройки

Для сброса на заводские настройки ....

### 3.5.2 Настройки пользователя.

Пользователь может задавать следующие параметры системы

1. Наименование Bluetooth точки
2. Пароль
3. Главный или ведомый (В цепи не может быть два главных устройства)
4. Количество устройств в цепи:
  - (a) Наименование точки
  - (b) Пароль точек
5. Количество датчиков
6. Тип выходных данных с датчика (клавиатура, мышь, геймпад)
7. Список возможных выходных команд:

```
[Команда 1:  
  [  
    [rx,ry,rz, ax,ay,z] ...  
  ],  
  ...  
]
```

При этом, если устройство является ведомым, то ему доступны "сложные" комбинации - комбинации которые состоят из двух или более цепей, тогда команда будет записываться в виде:

```
[Команда 1:  
  [  
    Номер устройства : Команда,
```

```

    ...,
    ],
    ...
]

```

## 3.6 Bluetooth

### 3.6.1 API

#### Считать данные с определенного датчика

Для ввода данной команды необходимо отправить g-[номер датчика], например взять данные с датчика 0: g-0. Выходные данные будут в следующей форме: [Время, [УголX, УголY, УголZ], [УскорениеX, УскорениеY, УскорениеZ]]. При этом время это время относительно старта работы устройства. Время передается в формате long long int, Углы и Ускорения в формате float

#### Считать данные со всех датчиков

Для ввода данной команды необходимо отправить g-а Выходные данные будут в следующей форме: [[Номер датчика, Время, [УголX, УголY, УголZ], [УскорениеX, УскорениеY, УскорениеZ]]...]. При этом время это время относительно старта работы устройства. Время и Номер датчика передается в формате long long int, Углы и Ускорения в формате float. Например:

```

[
    [0, 10, [0,14,32], [0,0,0]],
    [1, 13, [12,0,0], [10,0,0]]
]

```

### 3.6.2 Справка по командам

Для ввода данной команды необходимо отправить h. В качестве выходных данных будет строка неопределенной длины в которой будет содержаться справка о всех возможных командах

### 3.6.3 Конечный вывод команд

В качестве конечного вывода будут использованы HID команды ....