**Проект «роботизированная платформа**

**управляемая Arduino Uno и ESP32»**

**Ученика 10-а класса**

**Школы №29**

**Кочетков Л. С.**

**Руководитель: Горячев Владимир Алексеевич**

**2025**

Оглавление

[1. Введение 3](#_Toc188298103)

[4. Постановка задачи 4](#_Toc188298104)

[5. Анализ используемых технологий 5](#_Toc188298105)

[6. Описание конструкции и принципа работы 6](#_Toc188298106)

[8. Программная часть 9](#_Toc188298107)

9. Отрывок текста программного кода логики команд ESP32………….10

10. Отрывок текста программного кода логики платформы......................11

11. Тестирование и результаты…………………………………………………………..…….12

[12 Заключение 13](#_Toc188298108)

[13 Библиографический список литературы 14](#_Toc188298109)

1. **Введение**

Робототехника — это активно развивающаяся область науки и техники, которая внедряется во многие сферы жизни: от производства и складской логистики до медицины и бытовых устройств. Современные роботы могут выполнять сложные задачи, управляться удалённо или даже работать в автономном режиме.

Одним из наиболее востребованных направлений является дистанционное управление мобильными роботами. Такие устройства используются для исследования труднодоступных мест, перевозки грузов, взаимодействия с объектами и выполнения различных манипуляций.

В данном проекте реализуется **управляемая дистанционно мобильная платформа с манипулятором (клешнёй)**, управляемая жестами руки. В отличие от традиционных пультов или смартфонов, для управления используется **специальное носимое устройство** (перчатка с датчиками).

1. **Обоснование актуальности проекта**

В повседневной жизни и промышленности всё чаще применяются беспроводные системы управления, обеспечивающие удобство и свободу передвижения. Например, дистанционное управление используется в роботизированных пылесосах, экзоскелетах, медицинских манипуляторах, складских роботах и других устройствах.

Разработка собственной **простого и доступного** мобильного робота с дистанционным управлением позволяет:

* Изучить принципы работы современных робототехнических систем.
* Освоить основы программирования микроконтроллеров и беспроводной связи.
* Разработать устройство, которое может применяться для демонстраций, экспериментов и образовательных целей.

**Почему выбран именно этот подход?**  
Проект использует **Bluetooth** и **гироскопический датчик**, потому что:

* Bluetooth-модули обеспечивают **простую и надёжную связь** без необходимости сложных сетевых настроек.
* Использование **гироскопа (MPU6050)** позволяет управлять движением интуитивно, наклоняя руку в нужном направлении.

Такой способ управления удобен и понятен даже неподготовленному пользователю, так как он имитирует естественные движения человека.

**Какие задачи решает проект?**

* **Обеспечение интуитивного управления** — машинка движется в зависимости от наклона руки, без необходимости использования кнопок или джойстиков.
* **Организация беспроводного взаимодействия** — управление осуществляется по Bluetooth, без проводов и сложной настройки сети.
* **Практическое освоение робототехники** — работа с микроконтроллерами, датчиками и алгоритмами обработки данных.

1. **Цель и задачи проекта**

Цель проекта — разработка роботизированной системы дистанционного управления, основанной на использовании Bluetooth и инерциальных датчиков (MPU6050). Эта система должна обеспечивать интуитивное управление мобильной платформой (машинкой) и её захватным механизмом (клешнёй).

Для достижения цели были поставлены следующие задачи:

1. **Разработка механической части системы**
2. Спроектировать и собрать мобильную платформу (машинку), которая сможет перемещаться в разных направлениях.
3. Разработать и интегрировать захватный механизм (клешню) для манипуляций с объектами.
4. Создать управляющий блок в виде перчатки (или другого удобного носимого устройства).
5. **Создание программного обеспечения**

* Написать код для ESP32, который будет обрабатывать данные с датчиков и отправлять команды по Bluetooth.
* Разработать программную логику на Arduino Uno для обработки полученных команд и управления двигателями, сервоприводами и другими компонентами машинки.

1. **Тестирование и оптимизация работы системы**

* Проверить стабильность связи Bluetooth между устройствами.
* Оценить точность и быстродействие обработки команд от инерциального датчика.
* Оптимизировать программную часть для улучшения плавности управления.

1. **Постановка задачи**

**4.1 Описание задачи**

Современные робототехнические системы часто требуют сложных и дорогостоящих средств управления. В данном проекте разрабатывается **простой и удобный метод управления** мобильной платформой с помощью жестов руки.

Основная задача проекта: **создать роботизированную платформу, управляемую дистанционно при помощи наклонов руки и нажатия кнопки на управляющем блоке (перчатке).**

**4.2 Требования к системе**  
Для успешной реализации проекта были выдвинуты следующие требования:

* **Простота использования** – не требуется сложная калибровка или настройка. Достаточно включить оба устройства, после чего связь устанавливается автоматически.
* **Надёжность** – беспроводная связь должна быть устойчивой, а система должна работать без значительных задержек в передаче команд.
* **Функциональность** – машинка должна не только двигаться, но и управлять манипулятором (клешнёй).

1. **Анализ используемых технологий**
   1. **Обзор и выбор компонентов**

* **Модуль Arduino Uno:**Arduino Uno — это популярный микроконтроллер, использующий процессор ATmega328P. Он был выбран из-за:
* Простоты программирования
* Поддержки множества периферийных устройств
* Среднего количества памяти(32kB)
* Хорошего соотношения цена/функциональность

Arduino управляет двигателями машинки и сервоприводами клешни, обрабатывая команды, полученные по Bluetooth.

* **Модуль ESP32:**  
  Используется в управляющем блоке. Благодаря встроенному Bluetooth-модулю и высокой производительности, ESP32 идеально подходит для обработки данных от датчиков и их передачи на мобильную платформу.

ESP32 был выбран, потому что:

* Он имеет высокую производительность
* Имеет много различных протоколов связи
* Есть двухрежимный Bluetooth
* **Модуль HC-06:**  
  HC-06 — это простой Bluetooth-модуль, работающий в режиме "ведомого" (Slave). Он получает команды от ESP32 и передаёт их Arduino.

Этот модуль выбран, потому что:

* Он поддерживает последовательную передачу данных (UART), что облегчает интеграцию с Arduino.
* Работает на стандартной скорости 9600 бод, чего достаточно для передачи команд.
* Невысокая стоимость
* **Модуль MPU6050:**  
  MPU6050 — это комбинированный датчик, включающий **трёхосевой акселерометр и гироскоп**.

Акселерометр – показывает текущее ускорение по трём осям.

Гироскоп – показывает текущий угол наклона по трём осям.

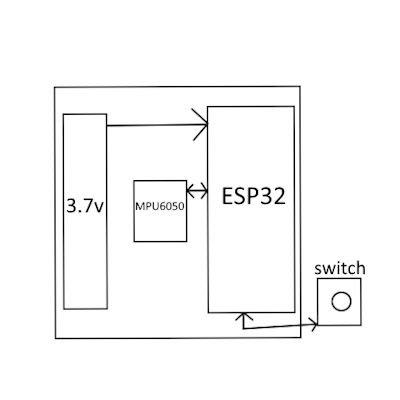
* **Сервоприводы:**  
  Небольшие устройства, модель mg90s, предназначенные для установки точного угла поворота вала. Управляются с помощью Широтно-Импульсной Модуляции (далее ШИМ).
  1. **Принцип работы канала связи Bluetooth**  
     Канал связи Bluetooth — это беспроводная технология, обеспечивает передачу данных на расстояние до 10 м. В проекте используется Bluetooth-канал для связи между управляющим блоком и мобильной платформой. ESP32 в управляющем блоке отправляет данные о состоянии датчиков и кнопки, а HC-06 на машинке принимает эти данные и интерпретирует команды для выполнения заданных действий.

1. **Описание конструкции и принципа работы**

Основой роботизированной системы является мобильная платформа, собранная на базе шасси от радиоуправляемой машинки.

* 1. **Расположение компонентов:**
  + **Захватный механизм (клешня):** установлен на передней части платформы и соединён с сервоприводами, которые обеспечивают сжатие и разжатие захвата.
  + **Центральный блок управления:**
    - **Arduino Uno** размещён в центральной части платформы. К нему подключены все исполнительные элементы: сервоприводы, моторы, а также модуль Bluetooth (HC-06).
    - Блок управления закреплён рядом с аккумуляторным отсеком.
  + **Аккумуляторный отсек:** находится в центральной части платформы и обеспечивает питание всех компонентов машинки (около 12 В).
  + **Силовая часть:** включает драйвер мотора и стабилизатор питания, расположенные в задней части платформы. Драйвер управляет моторами для движения платформы в двух направлениях (прямое и реверсное).
  1. **Описание конструкции управляющего блока**  
     Управляющий блок выполнен в виде компактного устройства, помещенного в эргономичный корпус, который надевается на тыльную сторону ладони пользователя.

**Расположение элементов управляющего блока:**

* + **Модуль ESP32:** основной микроконтроллер, закреплён внутри корпуса. Он обрабатывает данные от инерциального модуля (MPU6050) и кнопки, а затем отправляет команды машинке по Bluetooth.
  + **Модуль MPU6050:** установлен в центре коробки для считывания углов наклона.
  + **Кнопка:** выведена на внешнюю сторону корпуса и закреплена так, чтобы быть удобной для нажатия пальцем.
  + **Аккумулятор:** встроен внутрь корпуса блока и обеспечивает питание всех компонентов управляющего модуля.
  1. **Способ взаимодействия с платформой:**
* ESP32 считывает данные о наклонах и нажатии кнопки, преобразует их в команды: движение вперёд, движение назад, поворот руля (влево/вправо), управление клешнёй (открыть/закрыть) и отправляет команды по Bluetooth каналу связи.
* Модуль HC-06 на машинке принимает команды и передаёт их Arduino для выполнения.

**Структурная схема блока управления:**

Рис.1

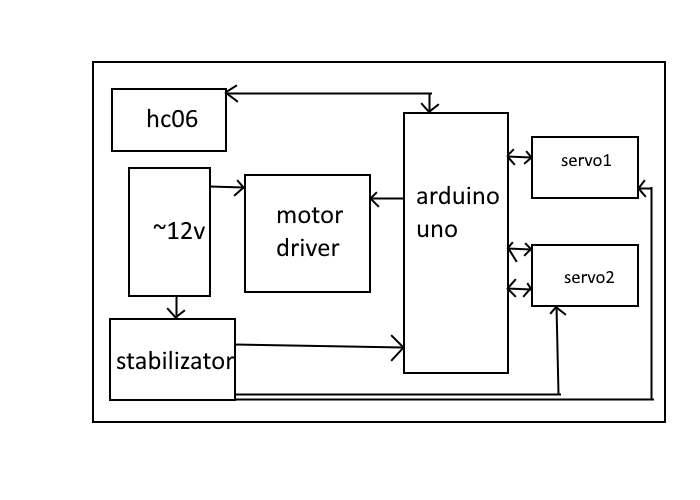
**Структурная схема мобильной платформы:**

Рис.2

1. **Принцип работы системы**
   1. **Последовательность действий пользователя:**
   * Включение платформы - пользователь переводит тумблер питания на корпусе машинки в положение "включено". После этого модуль HC-06 начинает ожидать подключения.
   * Включение управляющего блока (перчатки) - пользователь переводит тумблер питания на корпусе пульта управления в положение "включено". Перчатка автоматически подключается к модулю HC-06, после чего система готова к работе.
   * Управление:
     + Наклон блока вперёд/назад запускает движение машинки.
     + Наклон влево/вправо поворачивает рулевое управление.
     + Нажатие кнопки меняет положение клешни (сжатие или разжатие).
   1. **Передача и обработка данных:**

Данные с MPU6050 (углы наклона) и кнопки считываются ESP32 и преобразуются в команды.

* Команды отправляются на машинку через модуль Bluetooth.
* Модуль HC-06 принимает данные и передаёт их в модуль Arduino Uno, которая управляет моторами движения и сервоприводами захвата.

1. **Программная часть**

Описание логики программного обеспечения

**8.1 Работа с Bluetooth:**

* + Bluetooth модуль HC-06 принимает данные, отправленные ESP32.
  + ESP32 устанавливает связь с HC-06 при старте и регулярно отправляет команды, основанные на текущем состоянии датчиков и кнопки.

**8.2 Алгоритмы управления машинкой:**

* + **Движение:**  
    Arduino считывает команды от HC-06. В зависимости от них включаются/выключаются моторы, подключённые к драйверу, для движения вперёд/назад.
  + **Повороты:**  
    Сервопривод руля получает команды для установки заданного угла (влево, вправо, прямо).
  + **Управление клешнёй:**  
    Состояние кнопки отслеживается, чтобы переключать между командами сжатия и разжатия клешни.

**8.3 Обработка данных с MPU6050:**

* + ESP32 с помощью библиотеки для MPU6050 получает данные о текущих углах наклона.
  + Углы сравниваются с заданными пороговыми значениями, чтобы определить направление движения. Например:

1. **Отрывок текста программного кода логики команд ESP32:**

// Интегрируем данные гироскопа

    static *float* lastTime = millis() / 1000.0;

*float* currentTime = millis() / 1000.0;

*float* dt = currentTime - lastTime;

    lastTime = currentTime;

*float* gyroPitch = g.gyro.x \* dt;

*float* gyroRoll = g.gyro.y \* dt;

    // Комплементарный фильтр

    pitch = alpha \* (pitch + gyroPitch) + (1 - alpha) \* accelPitch;

    roll = alpha \* (roll + gyroRoll) + (1 - alpha) \* accelRoll;

    // Логика определения команды для движения

    if (roll >= 20 && roll <= 75) {

      movementCommand = 1; // Вперёд

    } else if (roll <= -20 && roll >= -75) {

      movementCommand = 3; // Назад

    } else {

      movementCommand = 0; // Стоп

    }

    // Логика определения команды для поворота

    if (pitch >= 20 && pitch <= 75) {

      steeringCommand = 6; // Вправо

    } else if (pitch <= -20 && pitch >= -75) {

      steeringCommand = 8; // Влево

    } else {

      steeringCommand = 0; // Стоп руля

    }

    // Логика отправки команд

    if (movementCommand != lastMovementCommand || steeringCommand != lastSteeringCommand) {

      String commandString = String(movementCommand) + "," + String(steeringCommand);

      SerialBT.println(commandString); // Отправляем команды через Bluetooth

      lastMovementCommand = movementCommand;

      lastSteeringCommand = steeringCommand;

    }

  }

1. **Отрывок текста программного кода логики платформы:**

// Обработка команд

*void* processCommands(String *receivedData*) {

*int* commaIndex = *receivedData*.indexOf(',');

  if (commaIndex != -1) {

    // Парсим команды движения и рулевого управления

    movementCommand = *receivedData*.substring(0, commaIndex).toInt();

    steeringCommand = *receivedData*.substring(commaIndex + 1).toInt();

    // Управление движением

    if (movementCommand == 1) {

      digitalWrite(move\_front, HIGH);

      digitalWrite(move\_back, LOW);

    } else if (movementCommand == 3) {

      digitalWrite(move\_front, LOW);

      digitalWrite(move\_back, HIGH);

    } else {

      digitalWrite(move\_front, LOW);

      digitalWrite(move\_back, LOW);

    }

    // Управление рулём

    if (steeringCommand == 8) {

      steer.write(steer\_right); // Вправо

    } else if (steeringCommand == 6) {

      steer.write(steer\_left); // Влево

    } else {

      steer.write(steer\_center); // Прямо

    }

  } else {

    // Парсим команду для клешни

    clawCommand = *receivedData*.toInt();

    if (clawCommand == 9) {

      claw1.write(claw\_close\_angle1);

      claw2.write(claw\_close\_angle2); // Закрыть клешню

    } else if (clawCommand == -9) {

      claw1.write(claw\_open\_angle1);

      claw2.write(claw\_open\_angle2); // Открыть клешню

    }

  }

}

1. **Тестирование и результаты**
   1. **Тестирование соединения Bluetooth:**

* На этапе проверки работоспособности связи между ESP32 и HC-06 было проведено тестирование подключения и передачи данных.
* Устройство успешно подключалось в течение нескольких секунд после включения обоих модулей.
* Тестирование включало проверку передачи команд при различных расстояниях. Максимальная дальность стабильной работы составила около 8–10 метров в помещении.
  1. **Проверка реакции машинки на команды:**
* Реакция машинки на команды (вперёд, назад, повороты) была протестирована при различных углах наклона перчатки.
* Машинка корректно интерпретировала команды, полученные через Bluetooth, и выполняла их с минимальной задержкой (~50 мс).
* Повороты руля происходили плавно, без рывков, при наклонах перчатки.
  1. **Тестирование работы клешни:**
* Клешня успешно переключалась между состояниями "сжата" и "разжата" при каждом нажатии кнопки на перчатке.
* Реакция системы на нажатие кнопки была мгновенной, а сервоприводы клешни работали стабильно без перегрева.
  1. **Соответствие реальной работы системе задачам:**
* Устройство полностью соответствует поставленным задачам. Управление машинкой происходит интуитивно, а Bluetooth-связь обеспечивает стабильное и надёжное соединение.
* Все основные функции, такие как движение, поворот руля и управление клешнёй, работают без сбоев.

### ****Заключение****

1. Цели проекта достигнуты: разработана роботизированная система управления, сочетающая простоту конструкции и функциональность.
2. Все задачи, поставленные в начале проекта, выполнены:
   * Спроектирована и собрана мобильная платформа.
   * Написано программное обеспечение для перчатки и машинки.
   * Проведено тестирование и оптимизация системы.
   1. **Перспективы развития проекта:**

* Добавление более сложной логики управления, например, автоматического удержания заданного направления или скорости.
* Улучшение конструкции клешни для захвата более тяжёлых или нестандартных объектов.
* Расширение функционала, например, внедрение датчиков для предотвращения столкновений.
  1. **Работа с компонентами:**
  + Подключение и настройка микроконтроллеров (Arduino Uno, ESP32).
  + Использование Bluetooth-модулей для связи.
  + Работа с инерциальным модулем MPU6050.
  1. **Создание программного обеспечения:**
  + Написание кода для взаимодействия с аппаратными модулями.
  + Реализация логики управления роботом.
  + Отладка и тестирование кода для бесперебойной работы системы.
  1. **Разработка робототехнических систем:**
  + Проектирование конструкции и компоновки компонентов.
  + Учет энергопотребления и работа с питанием.
  + Создание простой и интуитивной системы управления.

### ****Библиографический список литературы****

* 1. Документация на компоненты:
  + **ESP32-WROOM-32D:** https://docs.espressif.com/
  + **MPU6050:** https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/
  + **Arduino Uno:** <https://www.arduino.cc/>
  1. Учебные материалы:
  + "Программирование Arduino: основы" — Михаил Гончаров.
  + Онлайн-ресурсы по работе с Bluetooth и ESP32: Arduino Project Hub, Hackaday.