

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра электронных вычислительных машин
Дисциплина: Схемотехника

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему

БЕСПРОВОДНАЯ КОЛОНКА С УПРАВЛЕНИЕМ ЖЕСТАМИ

БГУИР КП 1-40 02 01 201 ПЗ

Студент: группы 150502
Альхимович Н.Г.

Руководитель: ассистент каф. ЭВМ
Стракович А.И.

Минск 2023

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет: КСиС. Кафедра: ЭВМ.

Специальность: 40 02 01 «Вычислительные машины, системы и сети».

Специализация: 400201-01 «Проектирование и применение локальных компьютерных сетей».

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

_____ Б.В. Никульшин
« ____ » _____ 2023 г.

ЗАДАНИЕ

по курсовому проекту студента
Альхимович Нины Геннадьевны

1 Тема проекта: «Беспроводная колонка с управлением жестами»

2 Срок сдачи студентом законченного проекта: 1 декабря 2023 г.

3 Исходные данные к проекту:

3.1 Микроконтроллер.

3.2 Определение жестов с помощью датчиков.

3.3 Динамики.

3.4 Источник питания.

4 Содержание пояснительной записки (перечень подлежащих разработке вопросов):

Введение 1. Обзор литературы. 2. Разработка структуры устройства. 3. Обоснование выбора узлов, элементов функциональной схемы устройства. 4. Разработка принципиальной электрической схемы устройства. 5. Разработка программного обеспечения. Заключение. Список использованных источников. Приложения.

5 Перечень графического материала (с точным указанием обязательных чертежей):

5.1 Беспроводная колонка с управлением жестами. Схема электрическая структурная.

5.2 Беспроводная колонка с управлением жестами. Схема электрическая функциональная.

5.3 Беспроводная колонка с управлением жестами. Схема электрическая принципиальная.

КАЛЕНДАРНЫЙ ПЛАН

Наименование этапов курсового проекта	Объем этапа, %	Срок выполнения этапа	Примечания
Обзор литературы	15	01.09 – 20.09	
Разработка структурной схемы	15	21.09 – 04.10	
Разработка функциональной схемы	20	05.10 – 23.10	
Разработка принципиальной схемы	15	24.10 – 05.11	
Разработка программного обеспечения	15	06.11 – 15.11	
Создание макета устройства	10	16.11 – 23.11	
Оформление пояснительной записки и графического материала	10	24.11 – 01.12	
Защита курсового проекта		07.12 – 19.12	

Дата выдачи задания: 14.09.2023 г.

Руководитель

А.И. Стракович

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 ОБЗОР ЛИТЕРАТУРЫ.....	6
1.1 Обзор аналогов	6
1.1.1 Sony LF-S50G Wireless Speaker.....	6
1.1.2 Bastron Bluetooth Speaker with hand gesture control	6
1.1.3 Gesture Control Bluetooth Speaker Arduino	7
1.2 Состав устройства	8
1.3 Микроконтроллеры.....	8
1.4 Динамики	8
1.5 Датчики движения	9
1.6 Датчики расстояния	10
2 РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА.....	11
2.1 Постановка задачи	11
2.2 Определение компонентов структуры устройства	11
2.3 Взаимодействие компонентов устройства	11
3 ОБОСНОВАНИЕ ВЫБОРА УЗЛОВ, ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА.....	13
3.1 Обоснование выбора микроконтроллера.....	13
3.2 Обоснование выбора динамиков	13
3.3 Обоснование выбора датчиков движения	13
3.4 Обоснование выбора датчиков расстояния	14
4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ УСТРОЙСТВА	15
4.1 Расчет мощности элементов схемы.....	15
4.2 Микроконтроллер	16
4.3 Динамики	16
4.4 Цифровой усилитель звука	16
4.5 Инфракрасные датчики движения.....	16
4.6 Ультразвуковой датчик расстояния	16
5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	17
5.1 Требования к разработке программного обеспечения	17
5.2 Блок-схема алгоритма.....	17
5.3 Исходный код программы.....	18
ЗАКЛЮЧЕНИЕ	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	20
Приложение А	21
Приложение Б.....	22
Приложение В	23
Приложение Г.....	24
Приложение Д	25
Приложение Е	32
Приложение Ж	33

ВВЕДЕНИЕ

Данный курсовой проект посвящен разработке микропроцессорного устройства на базе микроконтроллера. В качестве темы была выбрана беспроводная колонка с управлением жестами. Разрабатываемый проект посредством Bluetooth подключается к устройству (к примеру, смартфону), на котором будет выбираться и включаться необходимый аудиофайл для последующего воспроизведения. Контроль устройства будет осуществляться при помощи жестов, распознаваемых датчиками, такие операции, как:

- приостановку/возобновление прослушивания;
- перематывание композиций;
- изменение уровня громкости.

Хотя подобных беспроводных колонок на рынке представлено большое множество. В частности, лидирующие позиции в сфере занимают продукты таких компаний, как JBL, Marshall, Beats и пр. Подавляющим большинством товаров из обозначенной области, как правило, необходимо управлять через физическое взаимодействие с устройством.

Однако такой способ управления в определенного рода ситуациях может быть ограничен или неудобен. Собственно особенность и актуальность предлагаемой разработки заключается в том, что непосредственный контакт и расположение вплотную к колонке не требуется. Можно рассмотреть следующие примеры, которые отражают пользу описываемого устройства:

- если прямой подход к колонке затруднен или органы управления находятся вне зоны досягаемости пользователя;
- если определение положения управляющих кнопок или регуляторов производится владельцем устройства посредством органов зрения, а не осязания, то в ситуациях, требующих повышенной внимательности и концентрации, это может послужить отвлекающим фактором в отличие от предлагаемого способа;
- реализуемый метод взаимодействия позволит устранить проблему износа части механических деталей колонки, в частности кнопок, которая неизбежна в случае с традиционным подходом изготовления колонок.

Таким образом, разработка беспроводной колонки с управлением жестами нацелена на создание прототипа устройства, которое обеспечит более удобный и адаптированный под потребности пользователя функционал, а также, как курсовой проект, позволит получить навыки проектирования и реализации устройства.

1 ОБЗОР ЛИТЕРАТУРЫ

1.1 Обзор аналогов

1.1.1 Sony LF-S50G Wireless Speaker

Sony LF-S50G Wireless Speaker – это беспроводная колонка, которая подключается к сети Wi-Fi с целью управления, а также доступа ко всем «умным» устройствам. Она может воспроизводить музыку, контролировать совместимое домашнее оборудование, используя как голосовые команды посредством функции Google Assistant, так и бесконтактные жесты. Устройство изображено на рисунке 1.1.1.

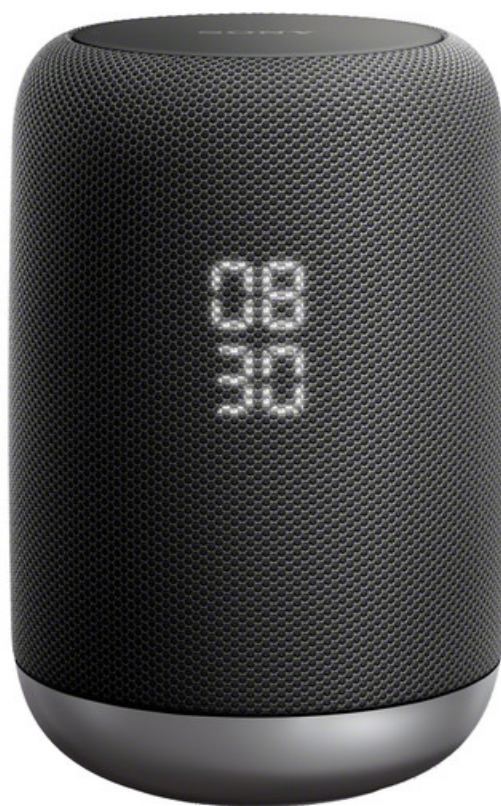


Рисунок 1.1.1 – Sony LF-S50G Wireless Speaker

1.1.2 Bastron Bluetooth Speaker with hand gesture control

Bastron Bluetooth Speaker – это уникальная беспроводная колонка, которая поддерживает управление жестами. Колонка имеет дизайн Magic Cube, в который не входит ни единая кнопка, изготовлена из полного алюминия и закаленного стекла. Встроенный аккумулятор на 2000 mAh обеспечивает длительное время работы. Колонка также оснащена встроенным микрофоном и функцией бесконтактного управления. Ее изображение приведено на рисунке 1.1.2.

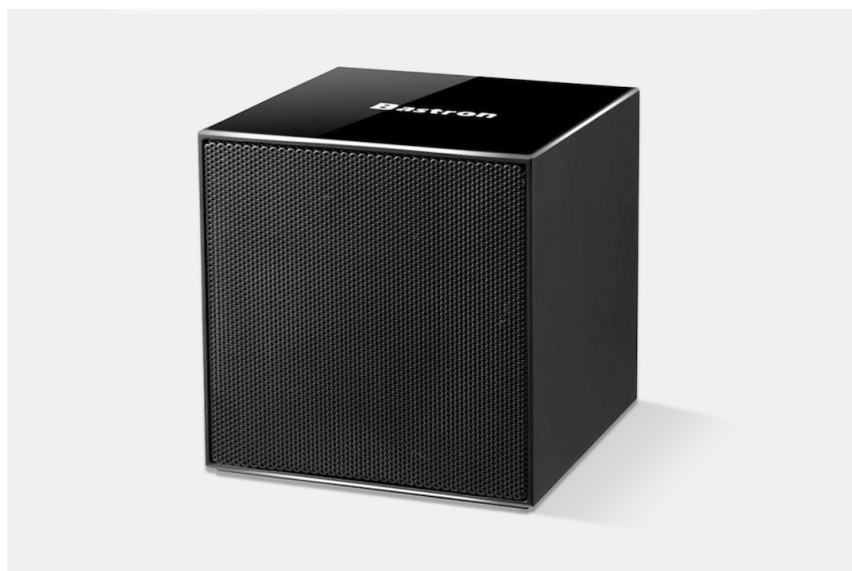


Рисунок 1.1.2 – Bastron Bluetooth Speaker

1.1.3 Gesture Control Bluetooth Speaker Arduino

Gesture Control Bluetooth Speaker Arduino – это Bluetooth-колонка, управляемая жестами, разработанная с использованием Arduino. Она представляет собой компактное устройство, которое позволяет пользователю управлять музыкой, просто проводя рукой над динамиком. Включение производится с помощью жеста взмаха рукой. При отсутствии использования производится автоматическое отключение. Колонка представлена на рисунке 1.1.3.



Рисунок 1.1.3 – Gesture Control Bluetooth Speaker Arduino

1.2 Состав устройства

В соответствии с описанным ранее, разрабатываемое микропроцессорное устройство выполняет функции воспроизведения музыки, которое управляется посредством жестов. Таким образом осуществляется приостановка и продолжение проигрывания, перемотка аудиозаписей и регуляция громкости звучания. Для реализации указанных задач устройство должно включать следующие компоненты:

- микроконтроллер;
- динамики;
- датчики движения;
- ультразвуковой датчик расстояния;
- аккумулятор;
- усилитель звука.

1.3 Микроконтроллеры

Микроконтроллеры предназначены для обработки информации и управления процессами обмена ею в составе микропроцессорной системы. Самыми широко используемыми моделями являются ESP32-WROOM-32 и Raspberry Pi 4, сравнение характеристик которых приведено в таблице 1.2.1.

Таблица 1.2.1 – Сравнение характеристик ESP32-WROOM-32 и Raspberry Pi

Параметр сравнения	ESP32-WROOM-32	Raspberry Pi 4
Вид процессора	Xtensa dual-core 32-bit LX6	Broadcom BCM2711
Процессор	от 80 МГц до 240 МГц	от 1,5 ГГц до 1,8 ГГц
Объем постоянной памяти	4/8/16 МБ	от 512 МБ до 32 ГБ
Объем оперативной памяти	520 КБ	от 256 МБ до 8 ГБ
GPIO	26	26
U (GPIO)	3,3 В	3,3 В
Аудиовыходы	Нет	HDMI, аудио разъем
Поддержка Bluetooth	Да	Да
Интерфейсы	Bluetooth, BLE, UART, SPI, SDIO, I2C	Bluetooth, BLE, USB

1.4 Динамики

Для воспроизведения звука необходимы динамики, коих существует достаточное количество аналогов. Для сравнения приведены продукты

компаний Aiyima и SOTAMIA в таблице 1.3.1. Их изображения приведены на рисунках 1.3.1 и 1.3.2.

Таблица 1.3.1 – Сравнение характеристик динамиков Aiyima и SOTAMIA

Параметр сравнения	Динамики AIYIMA	Динамики SOTAMIA
Количество каналов	1	2
Пиковая мощность	10 Вт	8-10 Вт
Источник питания	Нет	Нет
Суммарная мощность	до 25 Вт	до 25 Вт
Количество полос	Широкополосная	Широкополосная
Сопротивление	4 Ом	4/8 Ом



Рисунок 1.3.1 – Динамики AIYIMA



Рисунок 1.3.2 – Динамики SOTAMIA

1.5 Датчики движения

Датчики движения в общем случае реагируют на перемещение объектов и используются для контроля окружающей обстановки или автоматического запуска требуемых действий в ответ на перемещение объектов. В данном случае они будут применены для обнаружения жестов и их распознавания для дальнейших манипуляций воспроизведения. Такого рода датчики бывают различных видов: инфракрасные, ультразвуковые, радиоволновые, фотоэлектрические. В таблице 1.4.1 представлены характеристики

инфракрасного сенсора лучевого прерывания QT50CM и датчика движения HC-SR505.

Таблица 1.4.1 – Сравнение характеристик сенсора QT50CM и датчика HC-SR505

Параметр сравнения	QT50CM	HC-SR505
Рабочее расстояние	50 см	3 м
Напряжение питания	3,3 – 5,5 В	4,5 – 20 В
Время ответа	<2 мс	2-3 с
Рабочий диапазон температур	от -25°C до 60°C	от -20°C до +80°C

1.6 Датчики расстояния

Датчики расстояния, что исходит из названия, используются для измерения расстояния до объектов. В контексте данного проекта такой датчик необходим для манипуляций громкостью воспроизведения музыки. Из множества различных датчиков: ультразвуковых, инфракрасных, лазерных, емкостных – для сравнения были выбраны ультразвуковые датчики расстояния RCWL-1005 и HC-SR04. Их характеристики приведены в таблице 1.5.1.

Таблица 1.5.1 – Сравнение характеристик датчиков RCWL-1005 и HC-SR04

Параметр сравнения	RCWL-1005	HC-SR04
Рабочее расстояние	от 25 см до 4,5 м	от 2 см до 3,5 м
Рабочее напряжение	2,8 – 5,5 В	5 В
Потребляемый ток	2 мА	до 15 мА

2 РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА

2.1 Постановка задачи

В рамках данного курсового проекта требуется разработать беспроводную колонку с управлением жестами. Помимо соответствия стандартному назначению: воспроизведению музыки посредством Bluetooth-подключения к устройству, с которого будут включаться нужные композиции, будут реализованы следующие функции:

- остановка воспроизведения;
- продолжение воспроизведения;
- перемотка композиций в обе стороны, что будет распознаваться инфракрасными датчиками движения;
- увеличение или уменьшение громкости в зависимости от направления жеста и его амплитуды, что будет анализироваться ультразвуковым датчиком расстояния.

2.2 Определение компонентов структуры устройства

В соответствии с функциями, определенными в подразделе выше, определены компоненты, представленные ниже:

1. Микроконтроллер – главный элемент схемы, отвечающий за обработку поступающей с датчиков информации и отправку управляющих сигналов.
2. Динамики – для осуществления основной функции устройства – воспроизведения звука.
3. Модуль питания – источник питания схемы.
4. Датчики движения – датчики, которые улавливают производимые жесты и определяют их значение.
5. Датчик расстояния – датчик, который определяет размах производимого для управления громкостью жеста.
6. Модуль управления – кнопка для ручного управления.

Схема электрическая структурная, иллюстрирующая перечисленные компоненты и связи между ними приведена на чертеже ГУИР.400201.201 Э1 (приложение А).

2.3 Взаимодействие компонентов устройства

С помощью модуля управления в виде кнопки происходит перевод устройства в рабочее состояние. После чего для его дальнейшего использования необходимо посредством Bluetooth-соединения подключиться к устройству, с которого будет производиться включение воспроизведения.

Звук будет выводиться через динамики.

Когда будет происходить управление жестами (запрос на паузу или продолжение воспроизведения, перемотку), датчики движения будут анализировать полученные данные и в зависимости от вида, посылать тот или иной сигнал контроллеру, после чего будет осуществляться нужная пользователю манипуляция.

Если датчик расстояния зафиксирует вход производимого жеста в установленный диапазон, определенное значение будет передано микроконтроллеру и в зависимости от его величины должно произвестись соответствующее увеличение или уменьшение (зависит от направления жеста) громкости музыки.

Благодаря модулю питания осуществляется питание всех необходимых элементов схемы.

3 ОБОСНОВАНИЕ ВЫБОРА УЗЛОВ, ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА

3.1 Обоснование выбора микроконтроллера

Оба рассматриваемых микроконтроллера являются высокопроизводительными модулями, поддерживающими выполнение задач разного уровня сложности и нагрузки: от сетей маломощных датчиков до наиболее затратных задач, таких как звуковое кодирование, потоковое воспроизведение музыки, декодирование MP3.

В рамках разрабатываемого курсового проекта повышенные мощности не требуются, однако при выборе микроконтроллера учитывались такие параметры, как:

- поддержка Bluetooth, поскольку ее отсутствие потребовало бы включение в схему дополнительно Bluetooth-модуля;
- стоимость микроконтроллера.

Как уже было сказано выше, такие возможности в плане производительности, какие предлагает микроконтроллер Raspberry Pi 4, для данного устройства не требуются, а стоимость микроконтроллера серии ESP32 является более доступной для некоммерческих проектов, то выбор остановлен на микроконтроллере ESP32-WROOM-32, чьи характеристики удовлетворяют заданным целям. Также, стоит отметить, что выбранный модуль предлагает больший выбор сред разработки для написания программ, нежели продукт Raspberry.

3.2 Обоснование выбора динамиков

Главным требованием при выборе динамиков для будущего устройства является удовлетворительное качество звука, а также компактность и простота в подключении. Такие характеристики, как максимальная мощность и сопротивление находятся практически в одном диапазоне и обладающие ими динамики совместимы с выбранным микроконтроллером. Соответственно в проекте планируется использовать динамики AIYIMA, так как отзывы потребителей свидетельствуют о лучшем качестве звука, предоставляемом этим видом динамиков.

3.3 Обоснование выбора датчиков движения

На основании данных таблицы 1.4.1 предыдущего раздела рабочий диапазон датчика HC-SR505 представляется более удобным в использовании, чем первый аналог. Также, этот датчик обладает большим углом обнаружения (100 градусов против 10). Таким образом, в проекте будет применяться датчик HC-SR505 в том числе из-за доступности его приобретения.

3.4 Обоснование выбора датчиков расстояния

Хотя оба датчика могут быть использованы в целях, намеченных данным проектом, а именно: определении расстояния от руки человека до устройства, что будет использовано для определения желаемого изменения уровня громкости воспроизводимой музыки беспроводной колонкой; однако RCWL-1005 имеет более низкое рабочее напряжение и ток, а также поддерживает несколько режимов вывода. В то время как HC-SR04 имеет больший диапазон измерения и большую точность. Тем не менее, ключевым фактором, влияющим на окончательный выбор в сторону датчика HC-SR04, является наличие у датчика RCWL-1005 минимального диапазона в 25 см, что может создать трудности для работы устройства.

4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ УСТРОЙСТВА

4.1 Расчет мощности элементов схемы

Мощность, которую потребляет разрабатываемое устройство, складывается из мощностей, потребляемых его составными компонентами. Итоговая мощность элементов схемы устройства приведена в таблице 4.1.1.

Таблица 4.1.1 – Расчет мощности элементов схемы устройства

№	Блок	Количество	U, В	I, мА	P, мВт
1	Микроконтроллер ESP32-WROOM-32	1	3,3	240	792
2	Динамики AIYIMA	2	6,3	1580	10000
3	Датчик движения HC-SR505	2	5	0,06	0,6
4	Датчик расстояния HC-SR04	1	5	15	75
Суммарная мощность, мВт					867,6

Структуру схемы составляют:

- микроконтроллер ESP32-WROOM-32;
- 2 динамика AIYIMA;
- 2 датчика движения HC-SR505;
- датчик расстояния HC-SR04.

В результате, потребляемую мощность можно рассчитать следующим образом:

$$P_{\text{потр}} = I_1 \cdot U_1 + 2 \cdot I_2 \cdot U_2 + 2 \cdot I_3 \cdot U_3 + I_4 \cdot U_4 = 3,3 \cdot 240 + 2 \cdot 6,3 \cdot 1580 + 2 \cdot 5 \cdot 0,06 + 5 \cdot 15 = 20775,6 \text{ мВт} \quad (4.1.1)$$

Тогда максимальная потребляемая мощность с учетом поправочного коэффициента в 20%, который учитывает факторы, влияющие на реальное потребление электроэнергии, и позволяет скорректировать теоретический расчет максимальной потребляемой мощности для получения более точного значения:

$$P_{\text{потр}}^{\text{max}} = 1,2 \cdot P_{\text{потр}} = 29085,84 \text{ мВт} \quad (4.1.2)$$

Потребляемый ток равен следующему значению:

$$I_{\text{потр}} = \frac{P_{\text{потр}}}{U} = \frac{29085,84}{5} = 4617,168 \text{ мА} \approx 4,6 \text{ А} \quad (4.1.3)$$

4.2 Микроконтроллер

К нему подключаются все компоненты схемы через цифровые входы и выходы:

- к цифровым выходам (ЦАП) D25 и D26 подключен усилитель звука;
- к цифровым входам D2 и D15 – два датчика движения;
- к цифровым входам D4, D17 – датчик расстояния.

Данный микроконтроллер питается от напряжения 3,3 В.

4.3 Динамики

Микроконтроллер обеспечивает воспроизведение звука, связываясь с управляющим устройством посредством встроенной поддержки Bluetooth.

Хотя микроконтроллер допускает прямое подключение динамиков, однако воспроизводимый сигнал слабый по мощности.

Поэтому каждый из динамиков подключается к аналоговым выходам цифрового усилителя звука: Rout+, Rout- и Lout+, Lout- соответственно.

4.4 Цифровой усилитель звука

Подключение осуществляется следующим образом:

- VCC: подключается к BAT+;
- GND: подключается к земле;
- Rin: подключается к цифровому входу с ЦАП D25;
- Rout: подключается к цифровому входу с ЦАП D26 на ESP32.

4.5 Инфракрасные датчики движения

Подключение осуществляется следующим образом:

- VCC: подключается к +5В;
- GND: подключается к земле;
- OUT: подключается к цифровому входу D25 (левый сенсор) и D26 (правый сенсор) на ESP32.

4.6 Ультразвуковой датчик расстояния

Подключение осуществляется следующим образом:

- VCC: подключается к +5В;
- GND: подключается к земле;
- Trig (Trigger): подключается к цифровому входу D32;
- Echo: подключается к цифровому входу D33.

5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

5.1 Требования к разработке программного обеспечения

Разработанное устройство воспроизводит музыку. Управление режимом и параметрами проигрывания управляется посредством жестов.

Принцип работы устройства следующий. После подачи питания оно может начинать работу. Прежде всего, необходимо с помощью Bluetooth подключить источник звуковых данных к соответствующему модулю микроконтроллера. После выбора желаемой аудиозаписи динамики начнут ее воспроизводить.

В то же время датчики готовы к фиксации подаваемых жестов. За очередность проигрывания отвечают два инфракрасных датчика движения. Когда пользователь проводит рукой над устройством, анализируется состояние этих датчиков. Если сначала сработал левый, а затем через заданный интервал – правый, происходит переход к следующей композиции. Если наоборот сначала правый, а после – левый, то воспроизводится предыдущая аудиозапись.

Регулирование громкости производится при помощи ультразвукового датчика расстояния. Микроконтроллер побуждает его отправить ультразвуковой сигнал. Волна отражается от поверхности ладони и возвращается к датчику. Он обрабатывает время, прошедшее с момента генерации сигнала, и рассчитывает расстояние в сантиметрах.

Для того, чтобы два вида датчиков выполняли свои функции корректно, выбран диапазон расстояния и установлено граничное значение. Если рука прошла на расстоянии меньше, чем граничное, то происходит остановка и воспроизведение. Если жест производится выше установленного значения, то осуществляется регулирование громкости.

Если устройству не подается никакая управляющая информация, то оно работает в режиме обычного воспроизведения.

5.2 Блок-схема алгоритма

Блок-схемы алгоритма программного обеспечения приведена на чертеже ГУИР.400201.201 ПД (приложение Г).

Первые блоки отражают настройку необходимых параметров для дальнейшей работы программы. Далее начинается цикл, который не закончится до прекращения использования устройства. В нем происходит анализ информации, полученной при помощи датчиков, и определение того, какие жесты и на каком расстоянии от колонки выполнил пользователь. Последний параметр важен для функций остановки и воспроизведения, а также регулирования громкости. Производится его расчет. В последующих блоках формируются двоичные последовательности, на основании которых

будут описаны команды. Которые в свою очередь будут посланы на устройство, с которого выбираются аудиозаписи к прослушиванию.

5.3 Исходный код программы

Исходный код программного обеспечения приведен в приложении Д.

Функция `void setup()` реализует начальную настройку микроконтроллера: задает входные и выходные пины датчиков.

Функции `int getBeamL()` и `int getBeamR()` возвращают сигналы, пришедшие с датчиков движения.

Функция `void swiped()` анализирует состояние датчиков движения и фиксирует последовательность их срабатывания, присваивая соответствующее значение глобальной переменной `status`.

Функция `float distance()` реализует отправку сигнала датчику расстояния, измерение времени, затраченного на генерацию ультразвукового сигнала и отражения волны от поверхности руки, а также расчет расстояния, на котором рука находится относительно устройства.

Функция `void distanceToBits(int dist)` принимает в качестве аргумента `dist` рассчитанное в предыдущей функции значение расстояния и кодирует его в виде трехбитной последовательности.

Функция `Command bitsToCommand()` анализирует битовую последовательность и определяет, какую команду пользователь передал при помощи жеста.

Функция `int bitsToVolume(String binary)` принимает описанную битовую последовательность и возвращает уровень громкости, который требуется установить в качестве параметра воспроизводимой аудиозаписи.

Функция `void control(Command cmd, int vol)` осуществляет передачу полученной в качестве аргумента команды устройству, с которого производится выбор композиции.

ЗАКЛЮЧЕНИЕ

В результате работы над данным курсовым проектом было разработано микропроцессорное устройство, представляющее собой беспроводную колонку с управлением на основе жестов. Она выполняет базовые функции стандартного звуковоспроизводящего устройства, но предоставляет возможность бесконтактного взаимодействия с колонкой для регулирования очередности и громкости воспроизведения аудиозаписей.

В процессе реализации были получены практические навыки проектирования и разработки полноценного работоспособного устройства на базе микроконтроллера, а также закреплены принципы построения схем.

Среди достоинств данного проекта можно отметить:

- простота управления;
- удобство использования в случае затрудненного доступа к устройству;
- приемлемое сочетание цены и качества воспроизводимого звука.

В целях улучшения устройства в дальнейшем можно реализовать следующее:

- из-за несовершенства корпуса устройства звук может рассеиваться и терять качество, соответственно можно разработать более эргономичную и отвечающую требованиям компонентов проекта конструкцию;
- поскольку человеческая рука представляет собой неидеальную поверхность, которая имеет пропускающие участки, устройство не может обеспечить абсолютную надежность в контексте корректного срабатывания. Этот недостаток требует решения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Руководство для начинающих по ESP32 Bluetooth [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.electronicshub.org/esp32-bluetooth-tutorial/>

[2] Документация ESP32-WROOM-32 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

[3] Документация PAM8406 [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://pdf1.alldatasheet.com/datasheet-pdf/view/790839/DIODES/PAM8406DR.html>

[4] Как работает ультразвуковой датчик HC-SR04 и как его подключить к ESP32 [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://microkontroller.ru/esp32-projects/kak-rabotaet-ultrazvukovoj-datchik-hc-sr04-i-kak-ego-podklyuchit-k-esp32/>

[5] Работа с ИК-датчиком движения с использованием прерываний и таймеров на ESP32 [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://diyttech.ru/projects/rabota-s-ik-datchikom-dvizheniya-s-ispolzovaniem-preryvaniy-i-taymerov-na-esp32>

[6] Установка программного обеспечения [Электронный ресурс]. – Электронные данные. – Режим доступа: [https://developer.alexanderklimov.ru/arduino/arduino-setup.php#:~:text=Запускаем%20Arduino%20IDE%20и%20в,программу%20в%20Arduino\)%20в%20микроконтроллер.](https://developer.alexanderklimov.ru/arduino/arduino-setup.php#:~:text=Запускаем%20Arduino%20IDE%20и%20в,программу%20в%20Arduino)%20в%20микроконтроллер.)

[7] Документация Bluetooth A2DP API [Электронный ресурс]. – Электронные данные. – Режим доступа: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/bluetooth/esp_a2dp.html

[8] Вычислительные машины, системы и сети: дипломное проектирование (методическое пособие) [Электронный ресурс]: Минск БГУИР 2019. – Электронные данные. – Режим доступа: https://www.bsuir.by/m/12_100229_1_136308.pdf

Приложение А
(обязательное)

Схема электрическая структурная

Приложение Б
(обязательное)

Схема электрическая функциональная

Приложение В
(обязательное)

Схема электрическая принципиальная

Приложение Г
(обязательное)

Текст программы

Приложение Д (обязательное)

Код программы

```
#include "BluetoothA2DPSink.h"

#define BEAM_PIN_R 2
#define BEAM_PIN_L 15

//maximun volume distance in cm
#define MaxVolDist 30

//входы датчика расстояния
#define TRIGGER_PIN 4
#define ECHO_PIN 17

enum Command {
    DFLT = 0,
    PLAY,
    PAUSE,
    VOL,
    NEXT,
    PREVIOUS
};

//используются для кодирования команд
int S0_value = 0;
int S1_value = 0;
int S2_value = 0;
int S3_value = 0;
int S4_value = 0;

//переменные для хранения состояний плеера
int status = 0;
int playing = 1;

Command lastCmd = DFLT;

BluetoothA2DPSink a2dp_sink;
TaskHandle_t Task1;

void avrc_metadata_callback(uint8_t id, const uint8_t *text) {
    Serial.printf("==> AVRC metadata rsp: attribute id 0x%x, %s\n", id, text);
}

void setup() {
    const i2s_config_t i2s_config = {
        .mode = (i2s_mode_t) (I2S_MODE_MASTER | I2S_MODE_TX |
I2S_MODE_DAC_BUILT_IN),
```

```

        .sample_rate = 44100,
        .bits_per_sample = (i2s_bits_per_sample_t) 16,
        .channel_format = I2S_CHANNEL_FMT_RIGHT_LEFT,
        .communication_format =
(i2s_comm_format_t)I2S_COMM_FORMAT_STAND_MSB,
        .intr_alloc_flags = 0,
        .dma_buf_count = 8,
        .dma_buf_len = 64,
        .use_apll = false
    };

    a2dp_sink.set_i2s_config(i2s_config);

    Serial.begin(115200);
    a2dp_sink.set_avrc_metadata_callback(avrc_metadata_callback);

    //name
    a2dp_sink.start("GCBTSpeaker");

    pinMode(BEAM_PIN_L, INPUT_PULLUP);
    pinMode(BEAM_PIN_R, INPUT_PULLUP);

    pinMode(TRIGGER_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);

    Serial.println("Setup finished");
}

int getBeamL() {
    return digitalRead(BEAM_PIN_L) == HIGH;
}

int getBeamR() {
    return digitalRead(BEAM_PIN_R) == HIGH;
}

//считывание информации датчиками движения: сначала левый, потом
правый -> следующая песня; наоборот -> предыдущая
void swiped() {
    if ((getBeamL() == 1) && (getBeamR() == 0)){
        for (int counter = 0; counter < 1000000; counter++) {
            if (getBeamR() == 1) {
                Serial.println("*****Next Song*****");
                status = 1;
                break;
            }
        }
    }
    else if ((getBeamR() == 1) && (getBeamL() == 0)){
        for (int counter = 0; counter < 1000000; counter++) {
            if (getBeamL() == 1) {
                Serial.println("*****Previous Song*****");
            }
        }
    }
}

```

```

        status = 2;
        break;
    }
}
} else {
    status = 0;
}
}

float distance() {
    digitalWrite(TRIGGER_PIN, HIGH);    //отправить импульс
    датчику
    delayMicroseconds(10);
    digitalWrite(TRIGGER_PIN, LOW);

    unsigned long StartTime = micros();
    unsigned long StopTime = micros();

    while (digitalRead(ECHO_PIN) == 0) {
        StartTime = micros();
    }

    while (digitalRead(ECHO_PIN) == 1) {
        StopTime = micros();
    }

    float TimeElapsed = (float)(StopTime - StartTime);
    float distance = TimeElapsed * 0.0343 / 2.0;

    return distance;
}

void distanceToBits(int dist) {
    switch (dist) {
        case 0:
            S2_value = 0;
            S3_value = 0;
            S4_value = 0;
            break;
        case 1:
            S2_value = 1;
            S3_value = 0;
            S4_value = 0;
            break;
        case 2:
            S2_value = 0;
            S3_value = 1;
            S4_value = 0;
            break;
        case 3:
            S2_value = 1;
            S3_value = 1;

```

```

        S4_value = 0;
        break;
    case 4:
        S2_value = 0;
        S3_value = 0;
        S4_value = 1;
        break;
    case 5:
        S2_value = 1;
        S3_value = 0;
        S4_value = 1;
        break;
    case 6:
        S2_value = 0;
        S3_value = 1;
        S4_value = 1;
        break;
    case 7:
        S2_value = 1;
        S3_value = 1;
        S4_value = 1;
        break;
    }
}

void loop() {
    swiped();
    float dist = distance();

    //next song
    if (status == 1) {
        S0_value = 0; //0b00100 4
        S1_value = 0;
        S2_value = 1;
        S3_value = 0;
        S4_value = 0;
        loop2();
        delay(1000);
        status = 0;
        S2_value = 0;
        //previous song
    } else if (status == 2) {
        S0_value = 1; //0b00101 5
        S1_value = 0;
        S2_value = 1;
        S3_value = 0;
        S4_value = 0;
        loop2();
        delay(1000);
        status = 0;
        S0_value = 0;
        S2_value = 0;
    }
}

```

```

    } else if (dist <= (MaxVolDist + 4)) {
        if (dist < 4 && playing == 1) {
            Serial.println("PAUSE");
            playing = 0;
            S0_value = 0; //0b00010 2
            S1_value = 1;
            S2_value = 0;
            S3_value = 0;
            S4_value = 0;
            loop2();
            delay(1500);
            S1_value = 0;
        } else if (dist < 4 && playing == 0) {
            Serial.println("PLAY");
            playing = 1;
            S0_value = 1; //0b00001 1
            S1_value = 0;
            S2_value = 0;
            S3_value = 0;
            S4_value = 0;
            loop2();
            delay(1500);
            S0_value = 0;
        } else if (dist >= 4 && playing == 1) {
            Serial.println("DETECTING VOLUME");
            S0_value = 1; //0bxxx11 3
            S1_value = 1;
            int distScaled = int(((dist - 4) * 7 / (MaxVolDist -
4)));
            distanceToBits(distScaled);
        }
        //
        loop2();
        delay(200);
        S0_value = 0;
        S1_value = 0;
        S2_value = 0;
        S3_value = 0;
        S4_value = 0;
    }
} else {
    S0_value = 0;
    S1_value = 0;
    S2_value = 0;
    S3_value = 0;
    S4_value = 0;
}
delay(200);
}

Command bitsToCommand() {
    String res = ""; // "0b";

```

```

    res += S4_value ? "1" : "0";
    res += S3_value ? "1" : "0";
    res += S2_value ? "1" : "0";
    res += S1_value ? "1" : "0";
    res += S0_value ? "1" : "0";

    if (res[3] == '1' && res[4] == '1') {
        return VOL;
    } else {
        return static_cast<Command>(strtol(res.c_str(), 0, 2));
    }
}

int bitsToVolume(String binary) {
    String res = ""; //"0b";
    res += binary[2] == '1' ? "1" : "0";
    res += binary[3] == '1' ? "1" : "0";
    res += binary[4] == '1' ? "1" : "0";

    int scaledVolume = strtol(res.c_str(), 0, 2) * 127/7; //(16 *
    (res.toInt(2) + 1) - 1) / 2 + 63;

    return scaledVolume;
}

//передача команд на смартфон...
void control(Command cmd, int vol) {
    if (cmd == PLAY) {
        a2dp_sink.play();
        Serial.println("PLAYcmd");
    } else if (cmd == PAUSE) {
        a2dp_sink.pause();
        Serial.println("PAUSEcmd");
    } else if (cmd == NEXT) {
        a2dp_sink.next();
        Serial.println("NEXTcmd");
    } else if (cmd == PREVIOUS) {
        //to begin of son
        a2dp_sink.previous();
        delay(10);
        //and again for previous song
        a2dp_sink.previous();
        Serial.println("PREVIOUScmd");
    } else if (cmd == VOL) {
        a2dp_sink.set_volume(vol);
        Serial.print("Volume cmd: ");
        Serial.println(vol);
    } else if (cmd == DFLT) {
    }
}
}

```

```

void loop2() {
    Command cmd = bitsToCommand();

    if (cmd == VOL) {
        int vol = bitsToVolume("0b" + String(S4_value) +
String(S3_value)+ String(S2_value));
        control(cmd, vol);
    } else {
        if (cmd != lastCmd) {
            control(cmd, 0);
        }
    }
}

```

Приложение Е
(обязательное)

Перечень элементов

Приложение Ж
(обязательное)

Ведомость документов