**Районная учебно-исследовательская конференция**

**«Юность Пинежья»**

Направление: **Программирование и информационные технологии.**

**Управление нагрузкой с помощью голоса**

**Исследовательская работа**

Выполнена учеником 10 класса

МБОУ «Междуреченская СШ №6»

Абрамовым Русланом Леонидовичем

Научный руководитель – учитель

МБОУ «Междуреченской СШ №6»

Игнатьев Павел Алексеевич

**с. Карпогоры, 2022**

**Оглавление**

[Введение 3](#_Toc91162285)

[Описание устройства для управления нагрузкой 4](#_Toc91162286)

[Описание работы программы 5](#_Toc91162287)

[Заключение 7](#_Toc91162288)

[Библиографический список 8](#_Toc91162289)

[Приложение 1. Программа для работы устройства. 9](#_Toc91162293)

[Приложение 2. Внешний вид устройства. 10](#_Toc91162294)

# Введение

В настоящее время технология распознавания голоса является актуальной. Данную технологию широко применяют в быту. Пользователи могут управлять различными устройствами «умного» дома, такими как: система освещения, роботы пылесосы, газовые котлы и так далее. Сейчас на рынке стоит отметить устройствa таких компаний как Apple с их умной колонкой HomePod и Яндекс с Яндекс.Станцией[8]. В настоящее время используются два способа обработки голосовых данных: автономная и обработка на сервере. У каждого способа есть свои достоинства и недостатки. При обработке на сервере не требуется больших аппаратных возможностей, но с другой стороны нужно стабильное подключение к сети Интернет. При автономной обработке требуется специальное программно-аппаратное решения.

Цель - создать программно-аппаратный комплекс для управления нагрузкой с помощью голоса.

Объект исследования – программно-аппаратный комплекс.

В соответствии с данной целью были поставлены следующие задачи:

1. Создать и описать аппаратный комплекс.
2. Описать программу.

Методы исследования: теоретическое и практическое моделирование, эксперимент.

Гипотеза – с помощью голоса можно управлять нагрузкой.

# Описание устройства для управления нагрузкой

Устройство состоит из платы расширения EasyVR 3 Plus с микрофоном, платы контроллера Arduino Uno, силовое реле на 16А, блок питания на 5V. Всё оборудование смонтировано в корпусе настольной лампы. EasyVR позволяет записать перечень команд через микрофон, а потом определить, фразу с каким номером вы произносите [3]. Получив номер команды, микроконтроллер выполняет заложенное в неё действие. За обработку звука отвечает чип Sensory RSC-4128 выполненный по технологии «системе-на кристалле». Звуковой процессор объединяет в себе 8-битный микроконтроллер с интегрированным АЦП, ЦАП, ОЗУ, предусилителем и специализированными блоками обработки звука. Коннектор для подключение микрофона через разъём JST PH-2. Микрофон преобразует звуковые колебания в электрические, для последующей обработки звуковым процессором на модуле Easy VR Plus. Контроллер Arduino Uno  построен на ATmega328 [1]. Платформа имеет 14 цифровых вход/выходов (6 из которых могут использоваться как выходы ШИМ), 6 аналоговых входов, кварцевый генератор 16 МГц, разъем USB, силовой разъем, разъем ICSP и кнопку перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB, либо подать питание при помощи адаптера AC/DC или батареи. Реле для управления большой нагрузкой подключено к 13 аналоговому выходу контроллера

# Описание работы программы

Наша программа будет распознавать команды пользователя, и управлять нагрузкой в виде лампы. Мы будем использовать плату Arduino Uno как переходник USB→UART для начальной настройки EasyVR. Для этого установим на Arduino Uno шилд EasyVR, а затем поставим перемычку на Easy VR в положение PC [5]. Далее мы установили программу EasyVR Commander и подключили Arduino Uno к компьютеру при помощи USB-кабеля и в интерфейсе программы создали две команды и обучили их [6]. Для этого каждую команду мы произнесли дважды для проверки и присвоили им ID. ID 1 – команда ON(night), ID 2 – команда OFF(light). (Приложение 1)

Далее перейдем к программе контроллера, для этого мы на него поставили микроконтроллер. Опишем работу программы. В первой строке «#include <EasyVR.h>» мы подключаем библиотеку для соединения с платой EasyVR 3 Plus [2]. Строкой «EasyVR easyvr(Serial);» определяем параметры передачи данных через Serial-порт. Строкой «const int LED\_WAIT = 13;» мы называем номер пина для светодиода, индицирующего ожидание команды. Строкой «GROUP\_MAIN = 1;» мы указываем номер группы команд. Строкой «G1\_LIGHT = 0, G1\_NIGHT = 1;» мы прописываем команды из группы 1 занесённые в EasyVR Commander. Строкой «#define PIN\_RELAY 5» устанавливаем соответствие подключенному реле пятому аналоговому выходу. После строки «void setup(void)» мы устанавливаем скорость работы через Serial-порт. Далее строками «pinMode(LED\_WAIT, OUTPUT);…» переводим на запись порт для индикации активности микрофона. Arduinо передает данные на шилд EasyVR с помощью библиотеки. Ознакомившись со справочной информацией к библиотеке и плате расширения мы можем применить необходимые нам строки инициализации для ожидания соединения с платой, установки таймаута на распознавание и выбора языка. Это производится строками «while(!easyvr.detect()) delay(1000);» и далее. Процедура распознавания запускается строкой «void action(int8\_t group, int8\_t idx)»[7]. К этой процедуре мы будем обращаться из основного цикла программы. В ходе данной процедуры, когда найден оператор case, значение которого равно значению переменной,  выполняется программный код в этом операторе, то есть когда плата слышит команду «ON» выполняется подпрограмма строкой «case G1\_LIGHT:» и реле включается, а когда плата слышит команду «OFF» выполняется подпрограмма строкой «case G1\_NIGHT:» и реле выключается [4]. После строки «void loop(void)» начинается основная часть программы. После строки запуска функции «int8\_t idx» начинается работа светодиода по индикации режима платы, а если мы получаем код распознанной команды, то выполняем действие, а для этого переходим к выполнению процедуры описанной выше. Получаем код распознанной команды строкой «idx = easyvr.getCommand()», таким образом, если пользователь говорит команду «ON» модель лампы включается, а если «OFF» выключается.

# Заключение

В ходе проделанной работы была изучена и собрана плата EasyVR 3 Plus, уяснен принцип работы. После сборки устройства была написана программа для управления нагрузкой. Созданная программа позволяет управлять нагрузкой с помощью голоса. Устройство работает в автономном режиме, постоянно отслеживая параметры, используемые в программе. Наша программа использует процедуры и подпрограммы, плата Arduino Uno обменивается данными с платой EasyVR 3 Plus посредством библиотеки. Устройство имеет большое практическое значение. В дальнейшем программа может быть усложнена: увеличено количество команд, добавлено неточное распознавание и прочее.

# Библиографический список

1. Arduino Uno [Электронный ресурс] URL: http://arduino.ru/Hardware/ArduinoBoardUno/
2. Библиотека EasyVR Arduino [Электронный ресурс] URL: https://fortebit.tech/docs/manuals/easyvr-3/easyvr-arduino-library/

# Модуль EasyVR 3 Plus [Электронный ресурс] URL: http://wiki.amperka.ru/products:arduino-easyvr-3-plus-shield

1. Оператор выбора case Ардуино [Электронный ресурс] URL: https://робототехника18.рф/switch-case-arduino/
2. Преобразователь USB→UART [Электронный ресурс] URL: https://amperka.ru/product/troyka-usb-uart

# Программа EasyVR Commander [Электронный ресурс] URL: https://fortebit.tech/docs/manuals/easyvr-3/easyvrcommander/

1. Процедуры в Arduino [Электронный ресурс] URL: http://digitrode.ru/articles/3277-kak-sozdavat-i-ispolzovat-metody-funkcii-v-arduino-ide.html

## Яндекс Станция [Электронный ресурс] URL: https://yandex.ru/alice/station

# 

# Приложение 1. Программа для работы устройства.

#include <EasyVR.h>

EasyVR easyvr(Serial);

Servo\* srv;

const int LED\_WAIT = 13;

enum

{

GROUP\_MAIN = 1,

};

enum

{

G1\_LIGHT = 0,

G1\_NIGHT = 1,

};

enum

{

RELAY\_LIGHT,

RELAY\_NIGHT,

RELAY\_COUNT

};

enum

{

#define PIN\_RELAY 5

};

void setup(void)

{

Serial.begin(9600);

pinMode(LED\_WAIT, OUTPUT);

digitalWrite(LED\_WAIT, LOW);

pinMode(PIN\_RELAY, OUTPUT);

digitalWrite(PIN\_RELAY, HIGH);

while(!easyvr.detect()) delay(1000);

easyvr.setTimeout(5);

easyvr.setLanguage(EasyVR::ENGLISH);

}

void action(int8\_t group, int8\_t idx)

{

switch (group)

{

case GROUP\_MAIN:

switch (idx)

{

case G1\_LIGHT:

digitalWrite(PIN\_RELAY, LOW);

break;

case G1\_NIGHT:

digitalWrite(PIN\_RELAY, HIGH);

break;

}

break;

}

}

void loop(void)

{

int8\_t idx;

digitalWrite(LED\_WAIT, HIGH);

easyvr.recognizeCommand(GROUP\_MAIN);

while(!easyvr.hasFinished());

digitalWrite(LED\_WAIT, LOW);

delay(500);

idx = easyvr.getCommand();

if(idx >= 0)

{

action(GROUP\_MAIN, idx);

}

else

{

}

}

Приложение 2. Внешний вид устройства.

****