Министерство образования и науки РФ

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра Информационные технологии и автоматизированные системы

Современная технология Интернет вещей

Исследовательская работа

Тема: «Умный искатель ключей»

Выполняли: студенты группы АСУ-19БЗ

Слепченко Сергей и Аблаев Артур

Проверил: доцент кафедры ИТАС

Курушин Даниил Сергеевич

г. Пермь – 2021

Оглавление

[Цель работы 3](#_Toc92458824)

[Задачи работы 4](#_Toc92458825)

[Этапы выполнения 5](#_Toc92458826)

[Действующие элементы 5](#_Toc92458827)

[Принцип действия 6](#_Toc92458828)

[Программная реализация (код) 7](#_Toc92458829)

[Результаты 10](#_Toc92458830)

[Начальное состояние 10](#_Toc92458831)

[Замок в открытом состоянии 10](#_Toc92458832)

[Замок в закрытом состоянии 11](#_Toc92458833)

[Заключение 12](#_Toc92458834)

[Список используемой литературы 13](#_Toc92458835)

# Цель работы

Цель: создание виртуальной модели, имитирующей работу умного искателя ключей.

# Задачи работы

Задачи:

1. изучить принцип действия устройства;

2. создать программу для реализации работы умного искателя ключей на языке С++.

# Этапы выполнения

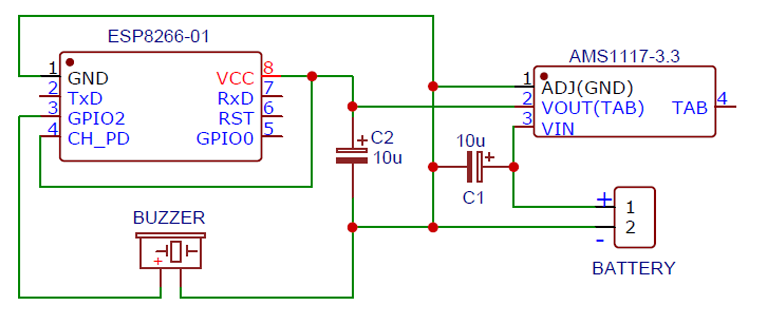
## Действующие элементы

Умный искатель ключей – устройство дистанционного действия, сигнализирующее о положении искомого объекта при отправке соответствующих запросов с управляемого устройства.

**Необходимые компоненты**

* Модуль ESP8266-01
* Регулятор напряжения 3.3V AMS1117
* Зуммер (Buzzer)
* Литий-полимерная батарея.
* Конденсатор 10 мкФ – 2 шт.

**Схема проекта**



## Разработка кода

1) Подключение библиотек, содержащих файлы для работы с платами ESP8266.

Библиотека ESP8266WiFi используется для подключения модуля ESP8266 к сети Wi-Fi.

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ESP8266WebServer.h>

2) Создадим объект веб-сервера, который будет принимать HTTP запросы на порту 80

ESP8266WebServer server(80);

3) Контакт модуля ESP8266, к которому подключен зуммер

const int buz\_pin = D2;

4) Параметры доступа сети Wi-Fi, к которой будем подключаться

char ssid[] = "Galaxy-M20"; //имя

char pass[] = "ac312124"; //пароль

5) HTML код для создания необходимой нам веб-страницы

String html\_code = "<!DOCTYPE html><html><head><style>.button {border: none;padding: 12px 40px;text-align: center;text-decoration: none;display: inline-block;font-size: 16px;margin: 4px 2px;cursor: pointer;}.button1 {background-color: black; color: white; border: 3px solid #1c1c1b;border-radius: 30px}body {text-align: center;}</style></head><body><h2>IoT Based Keychain</h2><p>Press the Button to Turn On/Off the Buzzer</p><form action=\"/BUZ\" method=\"POST\"><button class=\"button button1\">Click Me!</button></form>";

boolean buzzing\_state = false;

void handleRoot() {

server.send(200, "text/html", html\_code + "Current state: <b>" + buzzing\_state);

}

6) Функция handleRoot() будет исполняться когда мы будем открывать веб-страницу в браузере по адресу модуля ESP8266. На этой веб-странице будет отображаться состояние зуммера и кнопка для его включения.

void handleRoot() {

server.send(200, "text/html", html\_code + "Current state: <b>" + buzzing\_state);

}

7) Функция handleBUZ() будет использоваться для изменения состояния зуммера если на веб-странице нажали кнопку. Также эта функция добавляет в ответ браузеру новое положение (new location) чтобы в нем произошел возврат на главную страницу.

void handleBUZ() {

buzzing\_state = !buzzing\_state;

server.sendHeader("Location","/");

server.send(303);

}

8) Внутри функции setup() мы инициализируем последовательную связь со скоростью 9600 бод, зададим режим работы контакта, к которому подключен зуммер, на вывод данных и затем подключимся к Wi-Fi сети используя ее имя и пароль.

void handleNotFound(){

server.send(404, "text/plain", "404: Not found");

}

void setup(void){

  Serial.begin(115200);

  delay(10);

  pinMode(buz\_pin, OUTPUT);

  Serial.print("\n\nConnecting Wifi... ");

  WiFi.begin(ssid, pass);

  while (WiFi.status() != WL\_CONNECTED)

  {

    delay(500);

  }

  Serial.println("OK!");

  Serial.print("IP address: ");

  Serial.println(WiFi.localIP());

  Serial.println();

  server.on("/", HTTP\_GET, handleRoot);

  server.on("/BUZ", HTTP\_POST, handleBUZ);

  server.onNotFound(handleNotFound);

  server.begin();

  Serial.println("HTTP server started\n");

}

В следующих двух строчках кода функция 'handleRoot' вызывается когда пользователь запрашивает URI (Uniform Resource Identifier, унифицированный идентификатор ресурса, по сути компактная строка символов, используемая для идентификации абстрактного (виртуального) или физического сетевого ресурса. Формально определён в RFC 2396. URI, используемые в Web , именуются URL) "/". Вторая строка в представленном коде используется для вызова функции 'handleBUZ' когда запрос POST выполняется по URI "/ handleBUZ ".

server.on("/", HTTP\_GET, handleRoot);

server.on("/BUZ", HTTP\_POST, handleBUZ);

9) Далее мы будем считывать состояние зуммера с веб-страницы и затем соответствующим образом изменять состояние контакта, к которому подключен зуммер, чтобы его включать или выключать.

void loop(void){

  server.handleClient();

  if (buzzing\_state == true) {

    digitalWrite(buz\_pin, HIGH);

    delay(400);

    yield();

    digitalWrite(buz\_pin, LOW);

    delay(200);

    yield();

    delay(400);

    yield();

  }

}

## Программная реализация (код)

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ESP8266WebServer.h>

ESP8266WebServer server(80);

const int buz\_pin = D2;

char ssid[] = "Galaxy-M20";

char pass[] = "ac312124";

String html\_code = "<!DOCTYPE html><html><head><style>.button {border: none;padding: 12px 40px;text-align: center;text-decoration: none;display: inline-block;font-size: 16px;margin: 4px 2px;cursor: pointer;}.button1 {background-color: black; color: white; border: 3px solid #1c1c1b;border-radius: 30px}body {text-align: center;}</style></head><body><h2>IoT Based Keychain</h2><p>Press the Button to Turn On/Off the Buzzer</p><form action=\"/BUZ\" method=\"POST\"><button class=\"button button1\">Click Me!</button></form>";

boolean buzzing\_state = false;

void handleRoot() {

server.send(200, "text/html", html\_code + "Current state: <b>" + buzzing\_state);

}

void handleBUZ() {

buzzing\_state = !buzzing\_state;

server.sendHeader("Location","/");

server.send(303);

}

void handleNotFound(){

server.send(404, "text/plain", "404: Not found");

}

void setup(void){

  Serial.begin(115200);

  delay(10);

  pinMode(buz\_pin, OUTPUT);

  Serial.print("\n\nConnecting Wifi... ");

  WiFi.begin(ssid, pass);

  while (WiFi.status() != WL\_CONNECTED)

  {

    delay(500);

  }

  Serial.println("OK!");

  Serial.print("IP address: ");

  Serial.println(WiFi.localIP());

  Serial.println();

  server.on("/", HTTP\_GET, handleRoot);

  server.on("/BUZ", HTTP\_POST, handleBUZ);

  server.onNotFound(handleNotFound);

  server.begin();

  Serial.println("HTTP server started\n");

}

void loop(void){

  server.handleClient();

  if (buzzing\_state == true) {

    digitalWrite(buz\_pin, HIGH);

    delay(400);

    yield();

    digitalWrite(buz\_pin, LOW);

    delay(200);

    yield();

    delay(400);

    yield();

  }

}

# Заключение

В процессе выполнения данной исследовательской работы:

1. изучен принцип действия устройства;

2. создана программа для реализации работы умного искателя ключей на языке С++.

# Список используемой литературы

1. <https://ru.dhgate.com/product/smart-finder-key-whistle-sensors-sound-keychain/535323364.html>

2. <https://russian.alibaba.com/product-detail/anti-lost-device-bluetooth-security-tag-smart-finder-key-wireless-key-finder-bluetooth-4-0-personal-alarm-60208382550.html>

3.<https://ru.wikipedia.org/wiki/%D0%97%D1%83%D0%BC%D0%BC%D0%B5%D1%80>