МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ

**Отчет по лабораторной работе №2**

на тему:

«**Работа с NodeMCU**»

по дисциплине

«**Инженерия программного обеспечения для систем реального времени и Интернета вещей**»

Выполнил:

Студенты группы

КТбо2-8

Кочубей Д.С.

Ледерер П.А

Проверил:

Ассистент

Кафедры МОП ЭВМ

Усатова Д. В.

Оценка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Таганрог 2021

1. Задание

Тема 1. Охрана дома.

Нужно защитить дом от несанкционированного доступа. Для этого использовать датчик движения. Но так как датчик движения будет реагировать не только на людей, а еще и на пробегающих мимо животных, то нужно сделать дополнительные проверки. Датчик касания, который включит Вашу охранную систему. И дополнительная проверка для доступа в дом по магнитному ключу или магнитной карточке. Световыми и звуковыми эффектами оповестите хозяев либо о успешном доступе в дом ими самими, либо о незаконном доступе.

* Данные с входных датчиков отображаются в приложении и подается сигнал на выходные устройства. Пользователю в приложении отображаются сообщения о датчиках:
* О положении поставленной/снятой сигнализации;
* Сообщение о том, кто вошел по карточке/ключу;
* Произошло ли движение.

|  |  |
| --- | --- |
| **Датчики (сенсоры) для съема данных** | **Устройства управления / отображения** |
| 1. Датчик касания 2. HC-SR501 (PIR датчик) 3. RFID МОДУЛЬ RC522 13.56MHZ + КАРТА + БРЕЛОК | Пьезоизлучатель |

1. Ход работы

В процессе выполнения лабораторной работы, было выполнено поставленное задание (см. Рисунок 1, Рисунок 2).

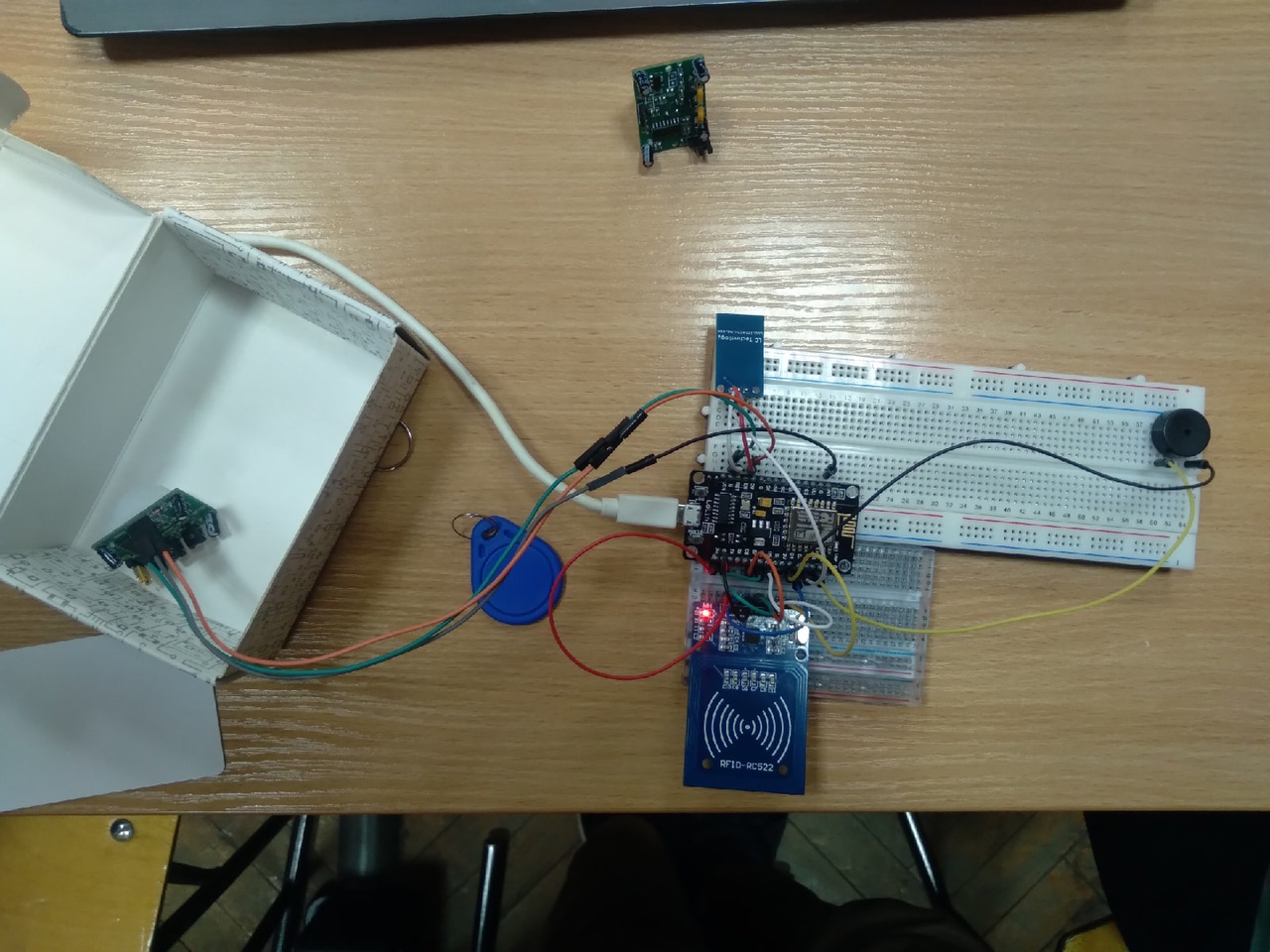


Рисунок — Общая схема подключения

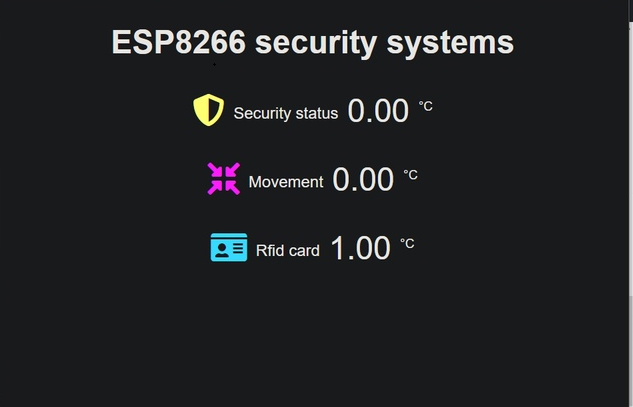


Рисунок — Информация на сайте

1. Листинг

#include <Arduino.h>

#include <ESP8266WiFi.h>

#include <Hash.h>

#include <ESPAsyncTCP.h>

#include <ESPAsyncWebServer.h>

// SPI setup

#include <SPI.h>

#include <MFRC522.h>

constexpr uint8\_t RST\_PIN = D3;

constexpr uint8\_t SS\_PIN = D4;

MFRC522 rfid(SS\_PIN, RST\_PIN); // Instance of the class

MFRC522::MIFARE\_Key key;

String tag;

int val; // Переменная для считывания состояния PIR

int inputPin = A0; // Пин подключения датчика

int ctsPin = 4; // пин для епкостного датчика касания

int piezoPin = 5; // Пин для пищалки

const char\* ssid = "example";

const char\* password = "password";

float s = 0.0;

float m = 0.0;

float r = 0.0;

float movement = 0;

// Create AsyncWebServer object on port 80

AsyncWebServer server(80);

const long interval = 10000;

const char index\_html[] PROGMEM = R"rawliteral(

<!DOCTYPE HTML>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" integrity="sha384-fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr" crossorigin="anonymous">

<style>

html {

font-family: Arial;

display: inline-block;

margin: 0px auto;

text-align: center;

}

h2 { font-size: 3.0rem; }

p { font-size: 3.0rem; }

.units { font-size: 1.2rem; }

.dht-labels{

font-size: 1.5rem;

vertical-align:middle;

padding-bottom: 15px;

}

</style>

</head>

<body>

<h2>ESP8266 security systems</h2>

<p>

<i class="fas fa-shield-alt" style="color:#808000;"></i>

<span class="dht-labels">Security status</span>

<span id="security\_status">%SECURITY\_STATUS%</span>

<sup class="units">&deg;C</sup>

</p>

<p>

<i class="fas fa-compress-arrows-alt" style="color:#FF00FF;"></i>

<span class="dht-labels">Movement</span>

<span id="movement">%MOVEMENT%</span>

<sup class="units">&deg;C</sup>

</p>

<p>

<i class="fas fa-id-card" style="color:#00add6;"></i>

<span class="dht-labels">Rfid card</span>

<span id="rfid\_card">%RFID\_CARD%</span>

<sup class="units">&deg;C</sup>

</p>

</body>

<script>

setInterval(function ( ) {

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200)

{

document.getElementById("security\_status").innerHTML = this.responseText;

}

};

xhttp.open("GET", "/security\_status", true);

xhttp.send();

}, 1000 ) ;

setInterval(function ( ) {

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200)

{

document.getElementById("movemen").innerHTML = this.responseText;

}

};

xhttp.open("GET", "/movemen", true);

xhttp.send();

}, 1000 ) ;

setInterval(function ( ) {

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200)

{

document.getElementById("rfid\_card").innerHTML = this.responseText;

}

};

xhttp.open("GET", "/rfid\_card", true);

xhttp.send();

}, 1000 ) ;

</script>

</html>

)rawliteral";

String processor(const String& var)

{

//Serial.println(var);

if(var == "SECURITY\_STATUS")

{

return String(s);

}

else if(var == "MOVEMENT")

{

return String(m);

}

else if(var == "RFID\_CARD")

{

return String(r);

}

return String();

}

void setup()

{

pinMode(inputPin,INPUT);

Serial.begin(115200);

SPI.begin(); // Init SPI bus

rfid.PCD\_Init(); // Init MFRC522

// Connect to Wi-Fi

WiFi.begin(ssid, password);

Serial.println("Connecting to WiFi");

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

{

delay(1000);

Serial.println(".");

}

// Print ESP8266 Local IP Address

Serial.println(WiFi.localIP());

server.on("/", HTTP\_GET, [](AsyncWebServerRequest \*request){

request->send\_P(200, "text/html", index\_html, processor);

});

server.on("/temperature", HTTP\_GET, [](AsyncWebServerRequest \*request){

request->send\_P(200, "text/plain", String(s).c\_str());

});

server.on("/humidity", HTTP\_GET, [](AsyncWebServerRequest \*request){

request->send\_P(200, "text/plain", String(m).c\_str());

});

server.on("/humidity", HTTP\_GET, [](AsyncWebServerRequest \*request){

request->send\_P(200, "text/plain", String(r).c\_str());

});

server.begin();

}

void loop()

{

// PIR block

movement = analogRead(inputPin);

// Serial.println(val);

delay(500);

if (movement > 100)

{

s = 1;

m = 1;

tone(piezoPin, 2000);

Serial.println("Зафиксированно движение!!");

// Button block

if (digitalRead(ctsPin)==HIGH)

{

Serial.println("Кнопка нажата!!!");

// RFID BLOCK

delay(500);

if ( ! rfid.PICC\_IsNewCardPresent())

return;

if (rfid.PICC\_ReadCardSerial())

{

for (byte i = 0; i < 4; i++)

{

tag += rfid.uid.uidByte[i];

}

}

Serial.println(tag);

if (tag == "14424717686")

{

noTone(piezoPin);

s = 0;

r = 1;

m = 0;

Serial.println("Правильный ключ!!");

movement = 0;

delay(10000);

}

if (tag == "221227218217")

{

r = 0;

Serial.println("Неправильный ключ!!");

}

tag = "";

rfid.PICC\_HaltA();

rfid.PCD\_StopCrypto1();

}

}

}