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)	П, спомолущетом	
3) RFID МОДУЛЬ RC522 13.56MHZ + KAPTA	Пьезоизлучатель	
+ БРЕЛОК		

2 Ход работы

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

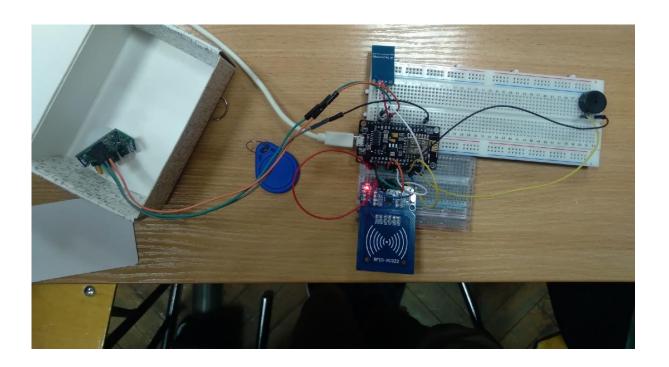


Рисунок 1 — Общая схема подключения



Рисунок 2 — Информация на сайте

3 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;
                    // Переменная для считывания состояния PIR
int val;
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-</pre>
scale=1">
      <link rel="stylesheet"</pre>
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>
      >
        <i class="fas fa-shield-alt" style="color:#808000;"></i></i></i>
        <span class="dht-labels">Security status</span>
        <span id="security status">%SECURITY STATUS%</span>
        <sup class="units">&deg;C</sup>
      >
```

```
<i class="fas fa-compress-arrows-alt"</pre>
style="color:#FF00FF;"></i>
        <span class="dht-labels">Movement</span>
        <span id="movement">%MOVEMENT%</span>
        <sup class="units">&deg;C</sup>
      >
        <i class="fas fa-id-card" style="color:#00add6;"></i></i>
        <span class="dht-labels">Rfid card</span>
        <span id="rfid card">%RFID CARD%</span>
        <sup class="units">&deg;C</sup>
      </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();
}
}
```