Тамара Тегелтија

**Паметни термометар**

ПРОЈКЕТКТНИ ЗАДАТАК

- Неиндустријска роботика и аутоматизација-

Нови Сад, септембар 2024.

САДРЖАЈ

[1. Увод 3](#_Toc177926115)

[2. Теоријске основе 6](#_Toc177926116)

[2.1. ESP8266 NodeMCU 6](#_Toc177926117)

[2.2. DHT11 сензор температуре и влаге 7](#_Toc177926118)

[2.3. Релејни модул 8](#_Toc177926119)

[3. Физичка реализација система 9](#_Toc177926120)

[3.1. DHT11 код и каблирање 10](#_Toc177926121)

[3.2. Вентилатор и диода код и каблирање 10](#_Toc177926122)

[3.3. ЕSP и веб интерфејс 12](#_Toc177926123)

[4. Закључак 14](#_Toc177926124)

# Увод

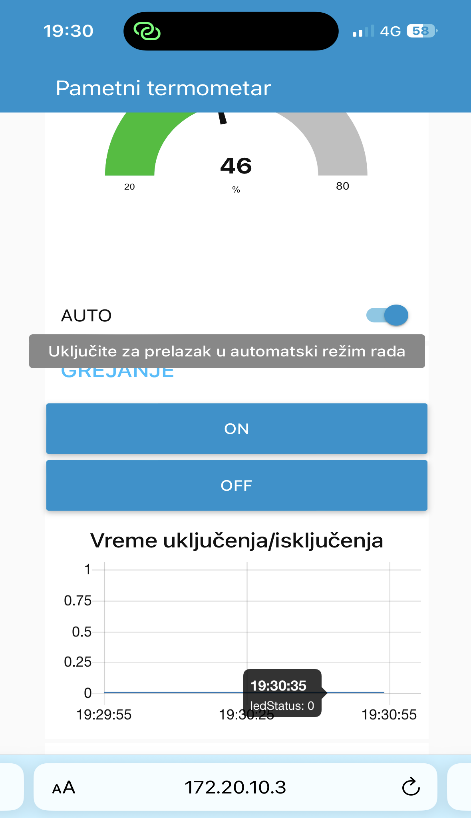
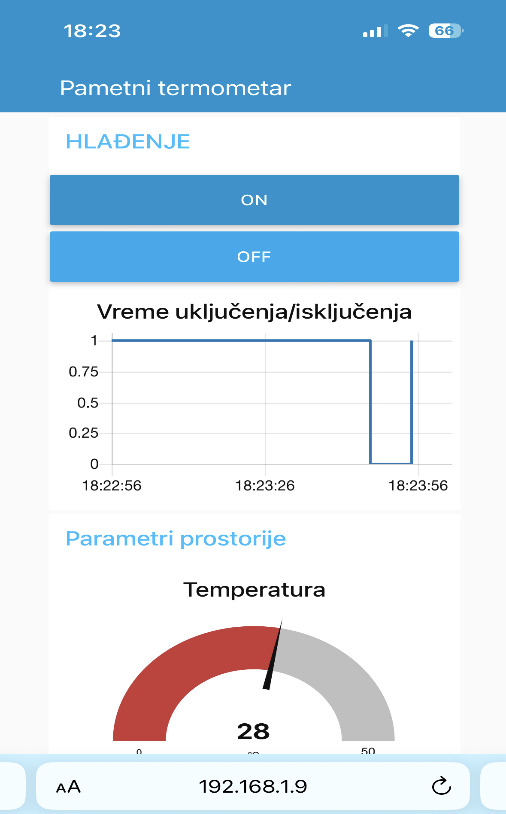
Главни циљ овог рада била је жеља да се упознам са свим аспектима развоја једног интегрисаног система на ESP8266 развојној платформи (која је постала веома популарна за развој IoT система). Мотивација је била да се направи систем који ће побољшати услове боравка у различитим просторијама, тако што неће бити потребе за физичким одласком до командне собе или до суседне собе у којој се налази клима или грејалица. Нема потребе за тражењем даљинског управљача, већ је то сада омогућено једним кликом на телефону који је у већини ситуација одмах поред нас.

Тема овог рада јесте реализација аутоматизованог система за терморегулацију који има задатак да у реалном времену очитава резултате саа сензора влаге и температуре и да у зависности у ком режиму рада се налази, да моментално дејствује.

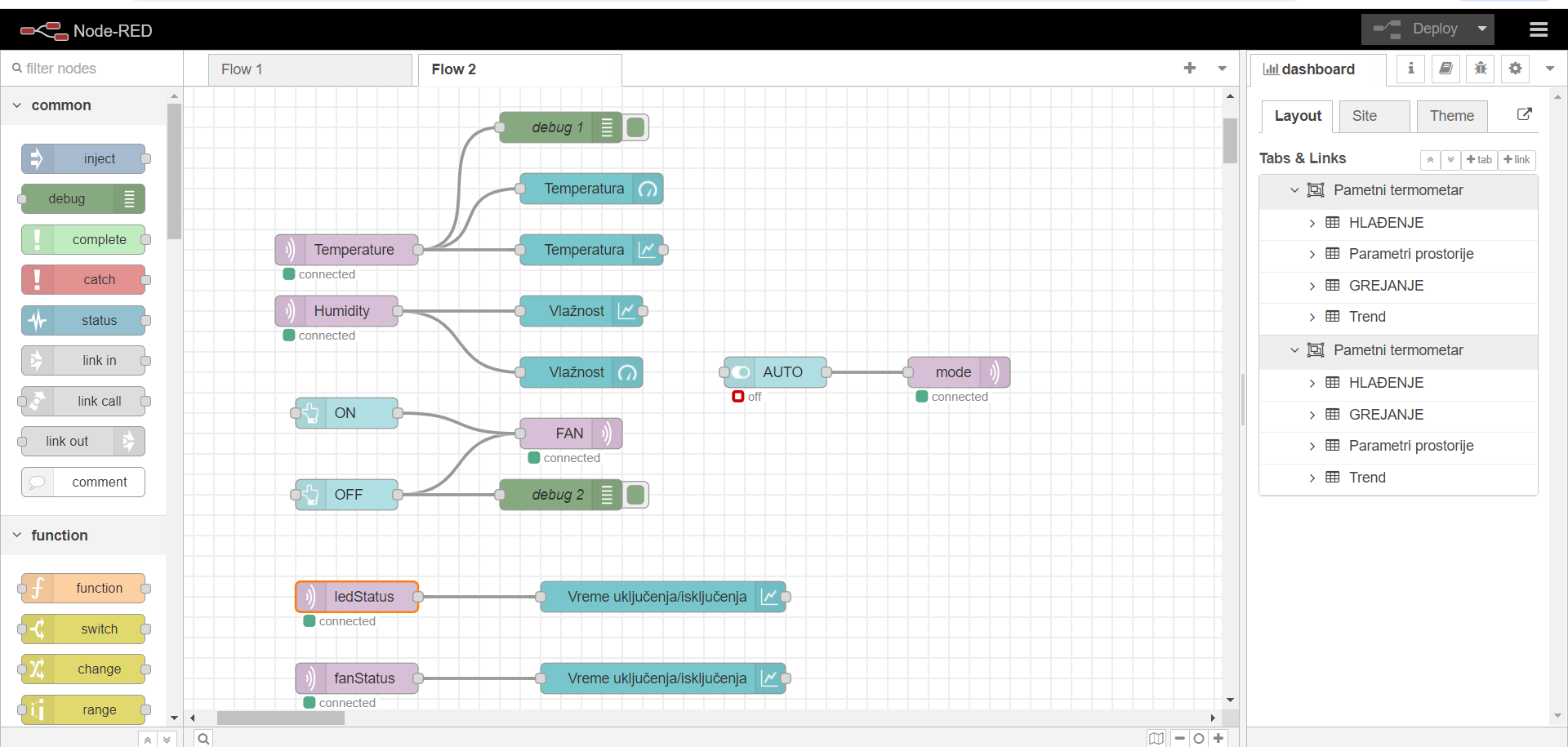
**Принцип рада:** ESP8266 NodeMCU детектује вредности са сензора DHT11 и проверава примљену поруку. Уколико су прекорачене граничне вредности и уколико је корисник подесио аутоматски режим рада, у зависности од температуре и влаге се укључује или вентилатор или грејалица. Пошто за израду макете није било могуће употребити ништа слично грејалици, за потребе симулације је одабрана црвена LED диода, па ће у даљем тексту грејалица бити називана диодом. Направљен је Node Red веб интерфејс са ког корисник може да очитава вредности температуре и влаге, да прати тренд промене тих вредности, као и да прати потрошњу енергије тако што временски може да очита колико су грејалица и клима били укључени. У ручном режиму рада може да контролише укључивање климе и грејалице у зависности од преференција, а да за то нема потребу да устаје, потребно је само да притисне одговарауће дугме у апликацији на телефону или персоналном рачунару. Такође, када није код куће систем може подесити да не ради у ручном, већ аутоматском режиму рада и тиме добија да када се врати увек има оптималну температуру просторија у којима се налази.



Слика 1: Изглед Node Red интерфејса на рачунару



Слика 2: Изглед Node Red интерфејса на мобилном телефону



Слика 3: Изглед окружења за прављење веб апликације

# Теоријске основе

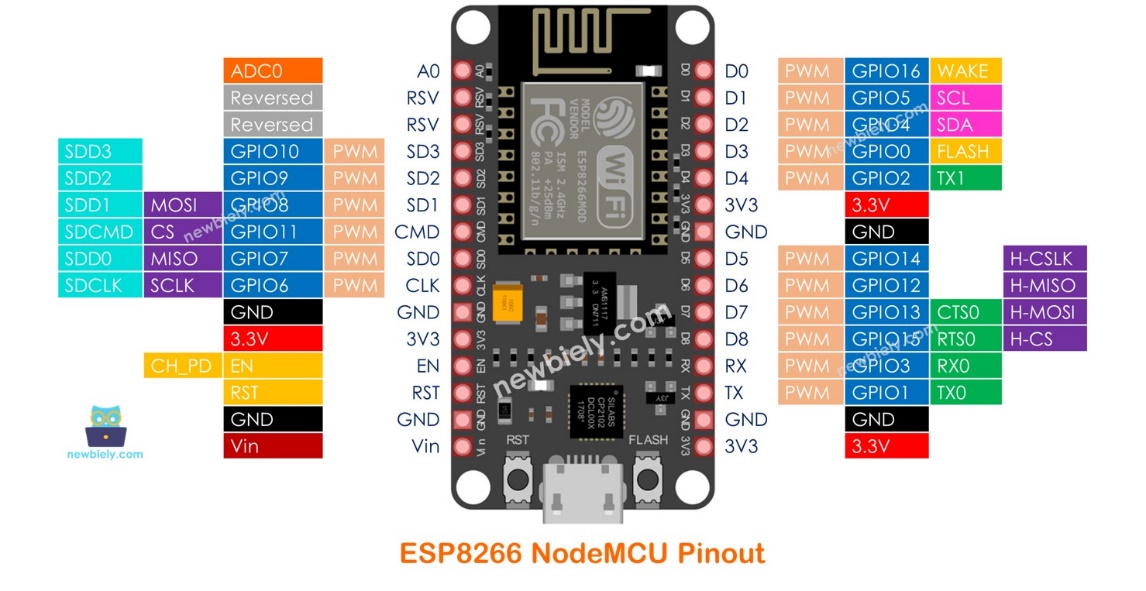
У наставку овог рада, биће описани принципи рада сваке од компоненти. Реализација самог рада заснована је на овладавању и разумевању конструкције и могућности које нам пружају ове сензорске, актуаторске и управљачке компоненте које су широко распрострањене и коришћене у различите индустријске и научне сврхе.

## ESP8266 NodeMCU

ESP8266 NodeMCU је микроконтролер намењен за Internet of Things (IoT) примене, произведен од стране компаније Espressif Systems. Због своје ниске цене, мале величине и прилагодљивости за embeded системе, ESP8266 се данас широко користи у IoT системима.

ЕSP8266 чип има Xtensa LX106 80 микропроцесор са једним језгром, има уграђену подршку за WiFi, али не и за Bluetooth, повезивање у радио опсегу од 160 MHz, 4MB флеш меморије за складиштење програма и података и 160KB SRAM за складиштење података. Такође садржи бројне периферије, укључујући тајмере и бројаче, 16 улазно-излазних пинова, један ADC конвертор, не садржи ниједан DAC конвертор. Могућност оспособљавања SPI, I2C, UART, I2S комуникације.

ESP8266 модул омогућава микроконтролерима да се повежу на Wi-Fi, користећи IEEE 802.11 bgn протокол. Може се користити са ESP-AT фирмвером за обезбеђивање Wi-Fi повезаности спољним микроконтролерима, или може функционисати као самосталан микроконтролер покретањем SDK-а заснованог на RTOS-у. Модул има пун TCP/IP стек и пружа могућност за обраду података, читање и контролу GPIO пинова.

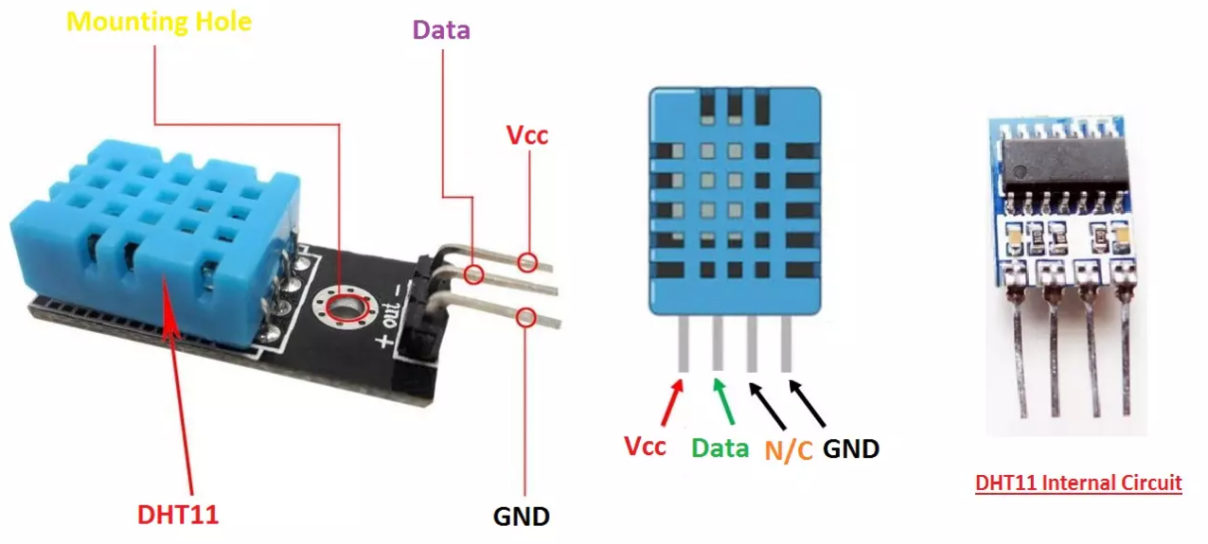
****

Слика 4: ESP8266 NodeMCU развојна плоча

## DHT11 сензор температуре и влаге

Овај сензор је фабрички калибрисан и није га потребно калибрисати, једина мана је што очитава вредности влаге и температуре на сваких секунду до две. Разлика између DHT11 и DH22 је што један мери вредности у опсегу од 0 до 50 степени док други мери вредности од -40 до +80 степени. Радни напон им је од 3-5V, DHT11 ради брзином од 1Hz док DHT22 брзином од 0,5 Hz. Такође DHT22 је скупљи, али и прецизнији. Оба раде на максимално 2,5mA. Код за ова два сензора се не мора мењати ако сe користи један, па се замени другим сензором.

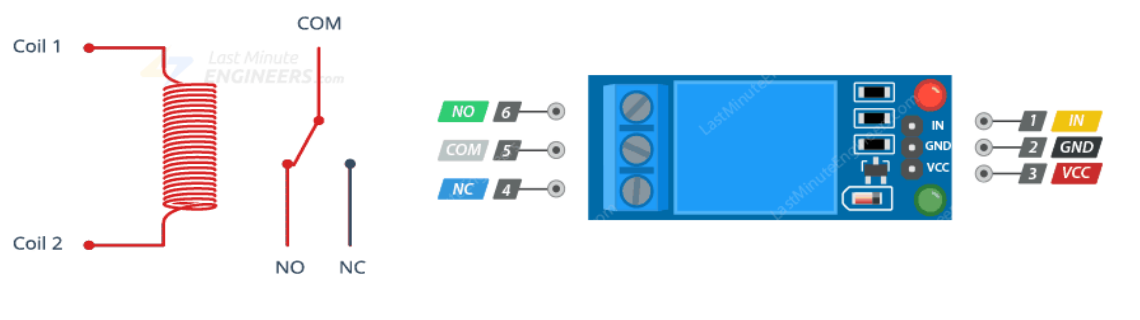
Сензор је сачињен од NTC термистора и компоненте која детектује влагу и има две електроде између којих се налази подлога која задржава влагу, обично со, или неки проводни пластични полимер. NTC термистор је врста отпорника чија се отпорност мења у зависности од промене температуре. За разлику од осталих отпорника овај тип отпорника је врло осетљив на промене температуре што га и чини погодним за примену у овој сензорској компоненти, промена отпорности му је око 100 ома по степену целзијуса. NTC значи да има негативни температурни коефицијент - отпорност се смањује порастом температуре.



Слика 5: DHT11 сензорски модул

## Релејни модул

Често се дешава да одређени кућни уређаји, лампе и вентилатори захтевају већи напон од оних 5V које можемо добити са Arduina. У случајевима када се јави потреба за већим радним напонима јавља се потреба за коришћењем електричних прекидача који у себи садрже електромагнет и активирају се побудним сигналом са микроконтролера. Ова компонента у себи садржи калем на који се доводи побудни напон са микроконтролера и када дође до успостављања магнетног поља мења се стање на прекидачу који може променити стање из нормално отвореног у нормално затворено или обрнуто у зависности од физичке реализације, пример је приказан на слици.



Слика 6: Релејни модул и његова конструкција

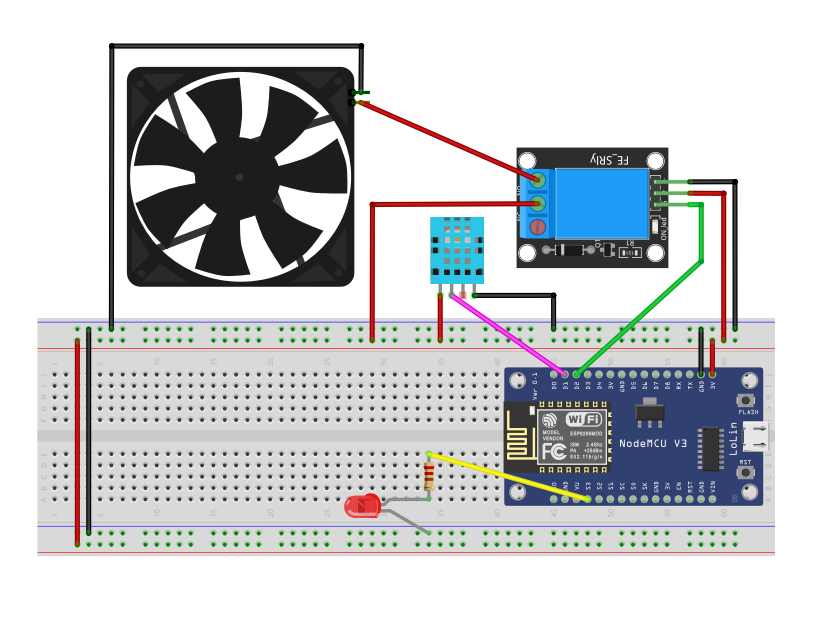
# Физичка реализација система

У наставку биће наведени и касније детаљно описани сви потребни кораци за реализацију овог рада као и шематски приказ на коме се могу видети све компоненте које су биле кориштене у овом раду:

1. DHT11 сензор, вентилатор и LED диоду с отпорником треба конектовати са ESP8266 развојном плочом
2. Направити веб интерфејс на ком ће бити приказане вредности са сензора и омогућити кориснику да преко веб интерфејса својевољно укључује и искључује

вентилатор и диоду

1. Када корисник на веб страници притисне дугме за аутоматски рад, биће очитане вредности са сензора и, у зависности од прага који је задат у програму, укључиваће/искључиваће се или диода или вентилатор.



Слика 7: Шематски приказ терморегулационог система реализованог помоћу ЕSP8266

## DHT11 код и каблирање

1. Да бисмо могли да радимо са овим сензором потребно је импортовати његову библиотеку у Arduino IDE окружењу.

#include <DHT.h>

1. Следећи корак је дефинисање константи:

#define DHTPIN 5 - избор пина који користимо за слање података

#define DHTTYPE DHT11 – бирамо тип сензора који користимо DHT11/DHT22/AM2302

DHT dht (DHTPIN, DHTTYPE); - DHT класа нам служи за повезивање са сензором

1. Потребне су нам две променљиве:

float humidity; - за складиштење вредности влаге

float temperature; - за складиштење вредности температуре

1. У void setup() функцији позивамо:

Serial.begin(9600); - иницијализација серијске комуникације, вредност у загради представља брзину комуникације и изражава се у битовима по минути

dht.begin(); - иницијализација DHT сензора

1. Следеће линијe кода које уносимо у void loop() су:

hum = dht.readHumidity(); - ишчитавање података и складиштење у променљивама

temp = dht.readTemperature();-ишчитавање података и складиштење у променљивама.

if (isnan(humidity) || isnan(temperature))

{Serial.println("Ne može da se pročita podatak sa DHT senzora!");

      return;} – заштита уколико сензор није успео да ишчита вредности температуре или влажности

## Вентилатор и диода код и каблирање

Део кода који је задужен за контролу вентилатора има задатак да у зависности од режима рада система (аутоматски или ручни) укључи или искључи вентилатор. У аутоматском режиму оног момента када је бар једна од вредности (температура или влажност) прекорачи граничне вредности које су предефинисане у коду. У овом режиму корисник нема могућност да сам управља вентилатором или диодом све док се не пребаци у ручни режим рада. Уколико се налази у ручном режиму рада, онда се све компоненте система активирају/деактивирају када корисник пожели. Потребно је нагласити да релејни модул ради у обрнутој логици и када се у коду пошаље високи сигнал, вентилатор се искључује и обрнуто. Такође, све поруке које ESP добија из Node Red-a (payload) су у ASCII запису карактера, па их је потребно у коду записати као децималне бројеве.

Објашњење кода:

#define FAN 4 – пин са ESP8266 који је повезан са IN командним пином са релеја  
 за паљење вентилатора.

#define LED 10 – пин са ESP8266 који користимо за диоду

bool manualMode = true; - променљива у којом је одређен режим рада, апликација је по дифолту у ручном режиму рада система на почетку

pinMode(FAN, OUTPUT); - подешавамо пин да се понаша као излазни (користимо га за   
 слање вредности), уносимо у void setup()

pinMode(LED, OUTPUT); - подешавамо пин да се понаша као излазни (користимо га за   
 слање вредности), уносимо у void setup()

digitalWrite(FAN, HIGH);- почетно стање пина се поставља на 0V, уносимо у

void setup()

if (payload[0] == 53) {  // Automatic mode

      manualMode = false;

      Serial.println("Switched to automatic mode");

    } else if (payload[0] == 54) {  // Manual mode

      manualMode = true;

      Serial.println("Switched to manual mode");} – део кода задужен за одређивање стања рада у ком се налази систем. Уколико корисник притисне дугме за аутоматски режим рада ESP прима поруку да је стигао број 5 и да се прелази у аутоматски режим, у супротном остаје порука са бројем 6 и систем је ручном режиму

if (manualMode) {

    if (String(topic) == "LED") {

      if (payload[0] == 49) digitalWrite(LED, HIGH);  // Payload 49 (turn LED on)

      else if (payload[0] == 50) digitalWrite(LED, LOW);  // Payload 50 (turn LED off)}

    if (String(topic) == "FAN") {

      if (payload[0] == 51) digitalWrite(FAN, HIGH);  // Payload 51 (turn FAN off)

      else if (payload[0] == 52) digitalWrite(FAN, LOW);  // Payload 52 (turn FAN on)}} – део кода када се систем налази у ручном режиму. Уколико је притиснуто дугме ОN у делу „ГРЕЈАЊЕ“ у интерфејсу добија се порука са бројем 1 и укључује се диода, а ако је притиснуто OFF добија се порука 2 и искључује се. Слично је и са вентилатором у делу „ХЛАЂЕЊЕ“, с тим да је порука за укључивање 3, а за искључивање 4

Овај део кода налази се у void loop() функцији и задужен је за аутоматски режим рада и проверу стања сензора:

if (!manualMode) {

    if (temperature > 26 || humidity>60) {

      Serial.println("Uključi FAN ON");

      digitalWrite(FAN, LOW);}

else {Serial.println("Isključi FAN OFF");

      digitalWrite(FAN, HIGH);} – Уколико је сензор измерио температуру већу од 25 степени Целзијуса или влажност већу од 60%, систем ће аутоматски да укључи хлађење, односно вентилатор. Уколико услов није испуњен, вентилатор се деактивира.

    if (temperature <= 24) {

      Serial.println("Uključi LED ON");

      digitalWrite(LED, HIGH);

    } else {

      Serial.println("Isključi LED OFF");

      digitalWrite(LED, LOW);}} – Уколико је сензор измерио температуру мању или једнаку 24 степена Целзијуса, потребно укључити грејање, односно у овом случају диоду, у супротном уколико услов није испуњен, потребно ју је искључити.

## ЕSP и веб интерфејс

1. Да бисмо почели са реализацијом слања и примања података са веб-а неопходно је успоставити комуникацију позивањем следећих библиотека:

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

1. Да би се ESP32 конектовао на мрежу потребно је унети име и шифру :

const char \*ssid = "RS";

const char \*pass = "piramida";

1. Потребно је подесити параметре за повезивање веб сервера на ком се налази интерфејса и ESP-a:

const char \*mqtt\_server = "broker.hivemq.com";

WiFiClient NodeMcuClient;

PubSubClient client(mqtt\_server, 1883, callback, NodeMcuClient);

Функција за очитавање теме:

void callback(char \*topic, byte \*payload, unsigned int length) {

Serial.print("Message arrived [");

Serial.print(topic);

Serial.println("]");

1. Прва ствар коју морамо написати у void setup() функцији је функција за почетак серијске комуникације:

Serial.begin(9600);- вредност у загради означава брзину комуникације

1. Следеће линије кода задужене су за успостављање комуникације са интернетом:

 Serial.println("Connecting to...");

WiFi.begin(ssid, pass);

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

    delay(500);

    Serial.print(".");}

Serial.println("WiFi Connected");

1. У void loop() функцији програмирамо оно што желимо да се деси када се клијент конектује на веб сервер. Уколико је конекција успешна, „субскрајбовањем“ или „паблишовањем“ је могуће посматрати податке система, као и управљари њиме бежично.

if (!client.connected()) {

    int Status = client.connect("224d305568e64c3384c4fba073927009.s1.eu.hivemq.cloud");

    if (Status) {

      Serial.println("MQTT Connected");

      client.subscribe("FAN");

      client.subscribe("LED");

      client.subscribe("mode");

    }

  }

bool ledStatus=digitalRead(LED);

    bool fanStatus=digitalRead(FAN);

    //publish-uj podatke sa senzora

    client.publish("Temperature", String(temperature).c\_str());

    client.publish("Humidity", String(humidity).c\_str());

    client.publish("ledStatus", String(ledStatus).c\_str());

    client.publish("fanStatus", String(fanStatus).c\_str());

# Закључак

У овом раду приказан је аутоматизовани терморегулациони систем реализован применом одговарајућег сензора, актуатора и ESP8266 развојне плоче. Предности овог терморегулационог система су постојање повратне спреге чиме је смањена потрошња електричне енергије када за то нема потребе и веб интерфејс помоћу којег корисници могу бити у интеракцији са системом и прилагођавати га сопственим потребама. Даљи развој овог рада може бити у смеру имплементације IAQ система (Indoor Air Quality System) у домаћинствима, зградама школама и другим радним просторијама и установама.