Универзитет Св. Кирил и Методиј
Факултет за информатички науки и компјутерско инженерство – Скопје

Имплементација на системи со слободен и отворен код

Тема:

Употреба на временско сериски бази на податоци во читање на информации од сензор

Изработил:

Стефан Димитровски
Индекс бр.181258

Ментор:

Проф. Д-р. Бобан Јоксимоски

Содржина

1. Абстракт

Во овој труд беше имплементиран систем со слободени и отворен код со употреба на временско сериска база на податоци, добиен со помош на соодветен сензор. Во изработката на проектот е користен ESP8266 , ниско буџетен WI-FI микрочип, InfluxDB OSS (open source software), DHT11 сензорот и Grafana веб апликација со отворен код за аналитика и интерактивна визуелизација на податоци.

Цел на проектот беше да се искористи микроконтролер заедно со сензор за температура и влажност. Добиените податоци од сензорот беа зачувани во временско сериска база и потоа истите ќе можат да бидат прикажани во графикон со цел да се анализираат за да се добијаат одредени заклучоци од мерењата.

2. Вовед

2.1. Важноста и потеклото на системи со отворен код

Лиценцата за отворен код поттикнува иновација преку соработка. Многу од системите кои имаат затворен код неможат јавно да се менуваат. Но по природа, отворен код овозможува системот да биде разгледан од голем број корисници кои побрзо би ги решиле проблемите за разлика од системите со затворен код.

Софтверот е дистрибуиран со неговиот изворен код со што тој е достапен за секој да го користи, изменува и дистрибуира.

Оваа идеја да се направи кодот отворен за сите поттекнува од 1983 со идеолошко движење неформално започнато од Richard Stallman (3), програмер на MIT кој верувал дека софтверот треба да биде достапен до секого, со цел да може да го разбере и да го подобри. Stallman започнал да го дистрибуира неговиот код за бесплатно под неговата лиценца GNU Public License (3). Ваквата нова идеологија придонесе до формирањето на Open Source Initiative во 1998 (3).

2.2. Микроконтролери и нивната корист

Микроконтролер е мал компјутер на единствен чип. Тој може да содржи едно или повеќе процесорски јадра, заедно со меморија и input/output периферали. Според нашите потреби ние можеме да го програмираме микроконтролерот да извршува одредени задачи. Ваквите компјутери се од голема важност во денешниот свет. Секој домашен уред е управуван од некаков вид микроконтролер, како машини за перење, клима уреди, далечински за телевизор и сл. Нивната примена уште повеќе расте во автомобилите, медицински уреди, канцелариска опрема, електрични алати, играчки, роботи и други интегрирани системи (Сл.1.). Постојат повеќе видови на микроконтролери кои при комбинација со одредени сензори ни овозможуваат да прибираме информации за околната средина и да ги обработуваме.

2.3. Временско сериски бази и нивната примена во индустријата

Временско сериска база е софтверски систем оптимизиран за чување и сервирање на временски серии преку парови од време и вредност (Сл.2). Често се користат алгоритми за компресија со цел ефикасно да се менаџира со податоците. Во дизајн на ваквите бази времето е клучен фактор со што значително е поразличен од релационите бази на податоци (4). Тие се користат во чување на податоци од Интернет на нештата, како мерења за вода, енергија и температура. Ваквите податоци континуирано треба да бидат проверувани со цел да се предвидат сезонски шеми, просечна употреба и неефикасности. Следно, може да се користат во проверка на перформансите за веб сервиси, апликации и инфраструктура. Понатаму за читање и разбирање на финансиски трендови, како и процесирање на податоци од автономни возила (5). Можностите со временско сериските бази секојдневно расте и се наоѓаат нови начини на читање и анализирање на податоците.

3. Изработка на проект користејќи временско сериска база и микроконтролер

3.1. Цел на проектот

Цел на проектот е да се искористи микроконтролер заедно со сензор за температура и влажност. Добиените податоци од сензорот ќе бидат зачувани во временско сериска база и потоа истите ќе можат да бидат прикажани во графикон со цел да се анализираат за да се стигне до одредени заклучоци од мерењата.

Како додаток, на микроконтролерот ќе работи веб сервер кој ќе се пристапува преку IP адресата на истиот уред, од каде ќе прикаже едноставна HTML страна со моменталната температура и влажност на воздухот. Вредностите ќе се ажурираат асихроно на одреден временски интервал.

3.2. Методологија

Во изработката на проектот ќе се користи ESP8266 , ниско буџетен WI-FI микрочип. Кој го подржува TCP/IP стекот. За временско сериска база ќе се користи InfluxDB OSS(open source software) која нуди API за внесување на податоци во базата преку TCP или UDP.

Микроконтролерот се програмира во Arduino IDE во C++ програмскиот јазик.

За прибирање на информации за температурата и влажноста на воздухот ќе се користи DHT11 сензорот, кој го мери околниот воздух и враќа дигитален сигнал на неговиот дигитален пин.

За да може ваквите информации да се превземат потребна ни е следната конфигурација (Сл.3., Сл.4).

3.3. Комуникација и зачувување на податоци во influxdb

Influxdata нудат клиент библиотеки со отворен код за конектирање до базата и за праќање на прашалници до неа. За тоа, прво се поставуваат соодветните параметри за WI-FI и базата (Сл.5), и го поврзуваме микроконтролерот (Сл.6).

Табелите во influxdb се нарекуваат кофички и за да се зачуваат податоци прво треба да се иницијализира објект point во кој ќе припаѓаат одредена класа мерења (Сл.8). Исто така може да се додаваат тагови со кои ќе ни покажат од која мрежа и уред доаѓаат мерењата. Вредностите што се запишуваат во базата се содржат од field и value парови, во нашиот случај првиот field ќе ни биде temperature со соодветните вредности и вториот field ќе ни биде humidity со неговите вредности измерени од сензорот (Сл.7).

3.4. Графички приказ на информациите

Додека influxdb нуди своја алатка за приказ на информациите зачувани во базата (Сл. 9), може да се користи Grafana, која претставува веб апликација со отворен код за аналитика и интерактивна визуелизација на податоци (Сл.10). За функционалност со апликацијата потребно е да се додаде influxdb на графана како data source и да се креира нова табла за визуелизација. Податоците се превземаат преку flux прашалници, специјален јазик подржан од influxdb.

Grafana има широка употреба и голема подршка за голем број апликации, со многубројни начини за прикажување на податоциите и опции за уредување на графиконите.

3.5. Креирање на веб сервер и асихроно ажурирање на податоците

Со симнување на ESPAsyncWebServer библиотеката и ESPAsyncTCP библиотека од github ни се овозможува да пуштиме веб сервер директно на ESP8266 микроконтролерот преку server.begin() функцијата (7). Користејќи ги методите на библиотеките може да се направи едноставна html страна која одредени вредности нотирани со %ИМЕ% ќе може да ги ажурираме асихроно на одреден интервал кој ние го дефинираме (Сл. 11, 12, 13).

4. Дискусија и проблеми при изработка на проектот

Како најпопуларна временско сериска база, Influxdb нуди голема колекција на алатки и сервиси со многу можности, но за нови корисници кои започнуваат со употреба на истите, може да биде поголем предизвик од очекуваното. Како секој софтвер со отворен код, зад него стои заедница која го одржува функционален и го развива. Заедницата на influxdb е помала за разлика од други алтернативи и затоа некогаш е тешко да се најдат одредени решенија за проблеми.

Документацијата може да биде збунувачка и поради големата разлика на верзиите од системот (1.7 и 2.0) често може да го пренасочи корисникот на дел од документацијата кој веќе не важи во поновата верзија.

Клиентските библиотеки за работа со базата се развивани од волонтери на заедницата и поради тоа не се без грешки. Конкретно, библиотеката за Arduino додека успешно функционира за работа со influxdb cloud, при работа со локалната верзија наидува на грешки за конекција.

Како што беше спомнато на почетокот, системите со отворен код се од голема важност. Истите ни овозможуваат да научиме за внатрешната функционалност на системите, да ги искористиме нашите веќе стекнати вештини, и можност да припаѓаме на заедница со исти интереси каде преку соработка овие системи можат да се усовршат и оптимизираат.

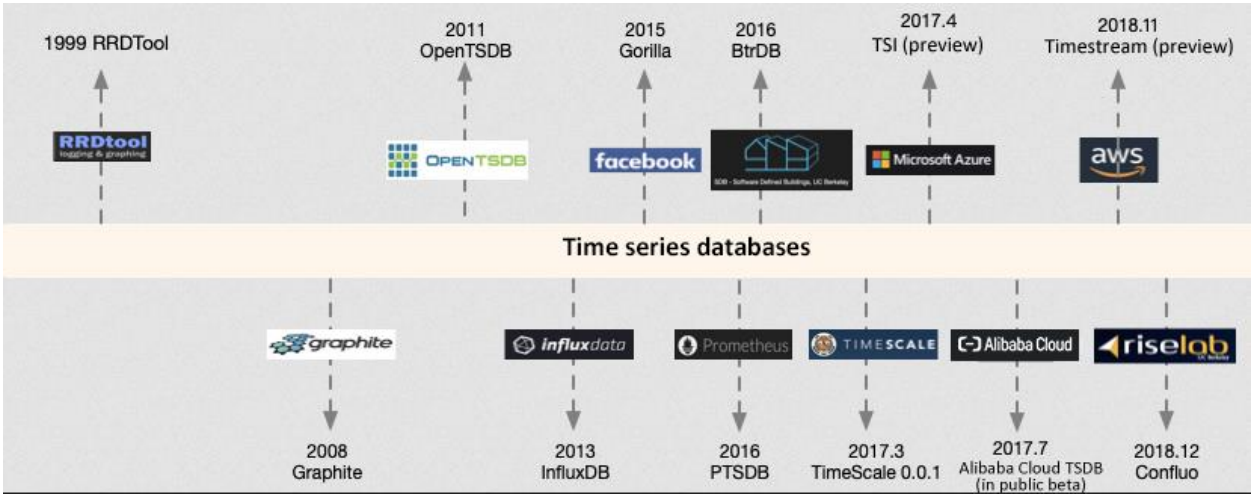
Лиценцата за отворен код заедно со системите за временски серии имаат значајна употреба во подобрување на нашиот бизнис преку следење на трендови, оптимизирање на нашите системи со промените во перформансите, предвидување на временски шеми од метеролошки мерења, предвидување на природни катастрофи, процесирање на сигнали, со еден збор можностите се - неограничени.

5. Литература

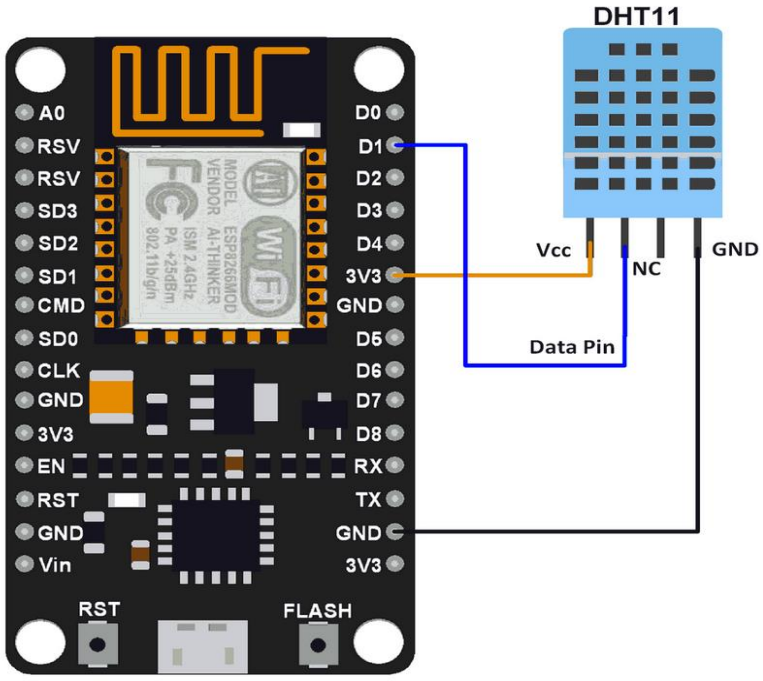
1. https://en.wikipedia.org/wiki/Open_source
2. https://lacewing.tech/blog/why-is-open-source-important/
3. https://www.synopsys.com/glossary/what-is-open-source-software.html
4. https://en.wikipedia.org/wiki/Time_series_database
5. https://aiven.io/blog/an-introduction-to-time-series-databases
6. https://en.wikipedia.org/wiki/Microcontroller
7. https://randomnerdtutorials.com/esp8266-dht11dht22-temperature-and-humidity-web-server-with-arduino-ide/

6. Додатоци

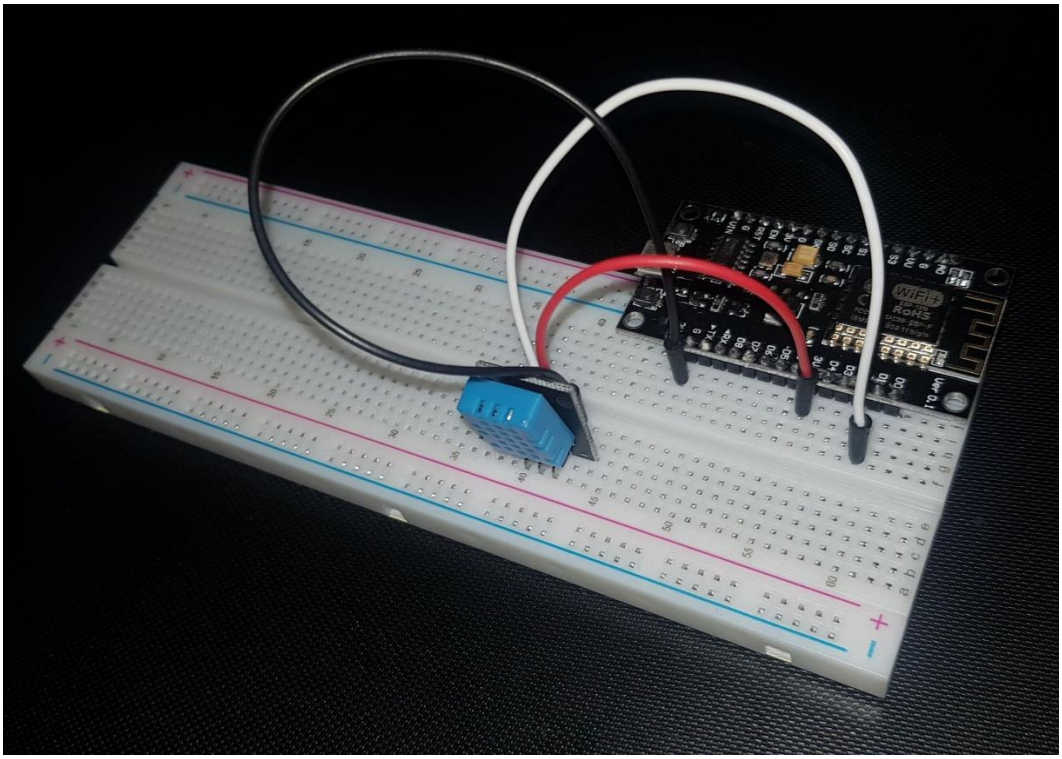
Слика 1. Примена на микрокнотролери во секојдневниот живот



Слика 2. Познати временско сериски бази на податоци



Слика 3. Поврзување на DHT11 со ESP8266



Слика 4. Пример поврзување

```
// Digital pin connected to the DHT sensor
#define DHTPIN 5
// Digital pin connected to the DHT sensor
#define DHTTYPE    DHT11
// WiFi AP SSID
#define WIFI_SSID "Stefan"
// WiFi password
#define WIFI_PASSWORD "12345678"
// InfluxDB v2 server url, e.g. https://eu-central-1-1.aws.cloud2.influxdata.com
#define INFLUXDB_URL "http://localhost:8086"
// InfluxDB v2 server or cloud API authentication token (Use: InfluxDB UI -> Data -> Tokens -> <select token>)
#define INFLUXDB_TOKEN "n4Hryy5CsNzFj6VFH886SHzSGLgJmCc5aoyrrxTWk7VtE3OW9e6cl9oF8gV6cZNlNMk6j6lrPzQ2bdUsr7ZZXA=="
// InfluxDB v2 organization id (Use: InfluxDB UI -> User -> About -> Common Ids )
#define INFLUXDB_ORG "stefan.dimitrovski_19@hotmail.com"
// InfluxDB v2 bucket name (Use: InfluxDB UI ->  Data -> Buckets)
#define INFLUXDB_BUCKET "test"
```

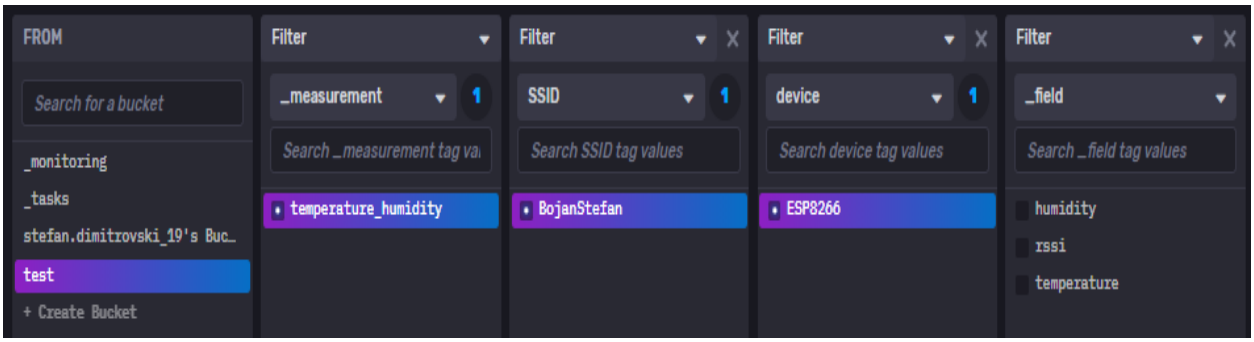
Слика 5. Параметри за WI-FI и базата

```
InfluxDBClient client(INFLUXDB_URL, INFLUXDB_ORG, INFLUXDB_BUCKET, INFLUXDB_TOKEN);

// Setup wifi
WiFi.mode(WIFI_STA);
wifiMulti.addAP(WIFI_SSID, WIFI_PASSWORD);

Serial.print("Connecting to wifi");
while (wifiMulti.run() != WL_CONNECTED) {
  Serial.println(".");
  delay(100);
}
Serial.println();

// Print ESP8266 Local IP Address
Serial.println(WiFi.localIP());
```

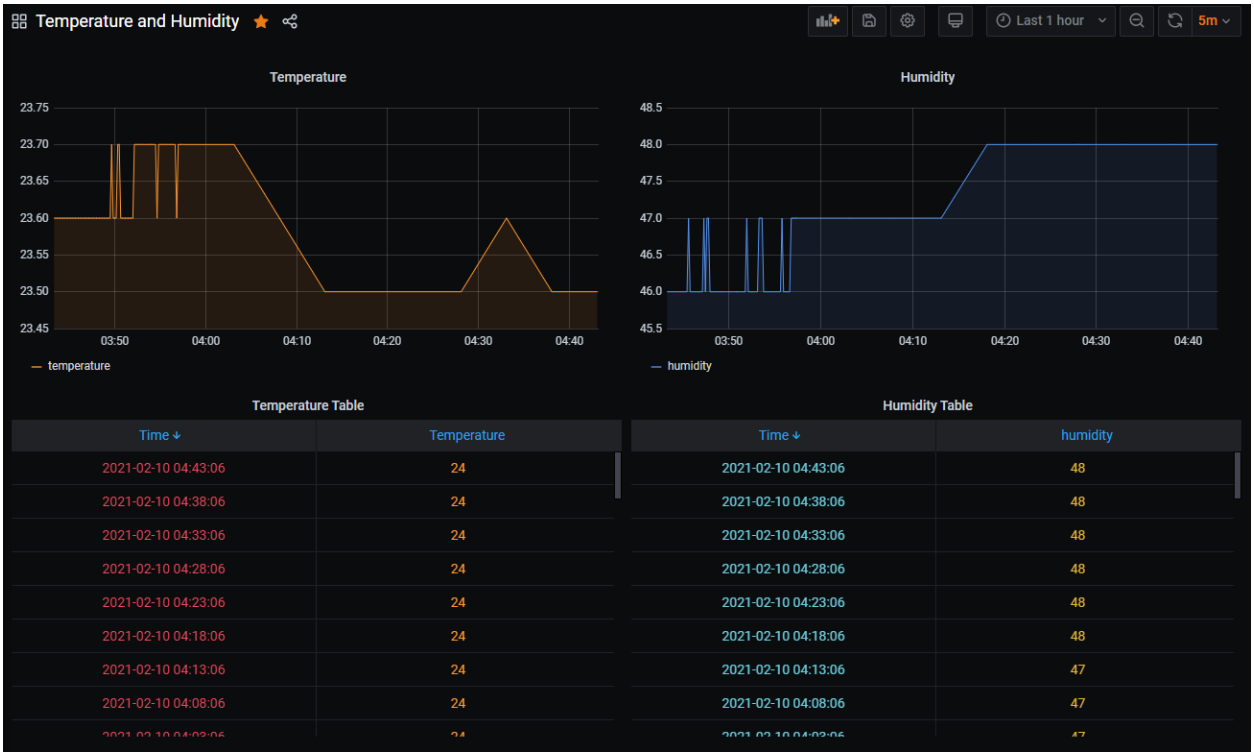
Слика 6. Конекција на WI-FI и базата

```
// Data point
Point sensor("temperature_humidity");

// Store measured value into point
// Report RSSI of currently connected network, current temperature and current humidity
sensor.addField("rssi", WiFi.RSSI());
sensor.addField("temperature", newT);
sensor.addField("humidity", newH);

// Print what are we exactly writing
Serial.print("Writing: ");
Serial.println(sensor.toLineProtocol());
```

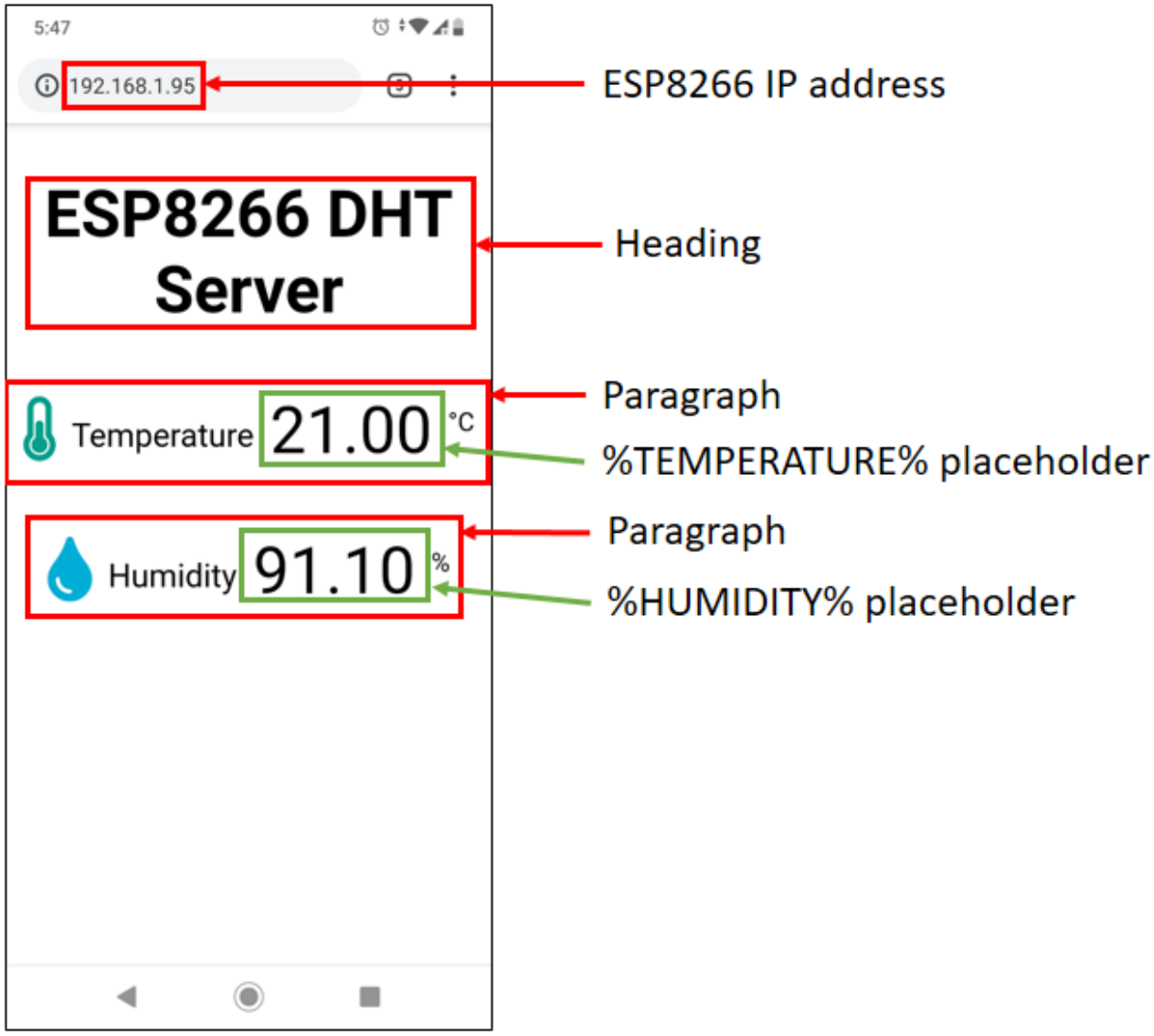
Слика 7. Зачувување на податоци во базата преку point објектот



Слика 8. Категоризирање на податоците во базата



Слика 10. Приказ на информациите преку Grafana



Слика 11. Изглед на HTML страната

```
setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("temperature").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "/temperature", true);
  xhttp.send();
}, 10000 ) ;
```

Слика 12. Javascript код за промена на вредноста на температура

```
unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= interval) {
  // save the last time you updated the DHT values
  previousMillis = currentMillis;
  // Read temperature as Celsius (the default)
  float newT = dht.readTemperature();
  // if temperature read failed, don't change t value
  if (isnan(newT)) {
    Serial.println("Failed to read from DHT sensor!");
  }
  else {
    t = newT;
    Serial.println(t);
  }
```

Слика 13. Читање вредноста на температурата од сензорот