NACKADEMIN

Examensarbete

Martin Myrberg

IOT-22

Examensarbete

Badvattentemperatur

- En IoT-lösning för vattentemperaturmätning

Student: Martin Myrberg

Utbildning och klass: IOT-22

Handledare: Oscar Bexell

Datum och ort: 2024-01-11, Stockholm

NACKADEMIN

Sammanfattning

Projektet *”En IoT-lösning för vattentemperaturmätning"* syftar till att utveckla en IoT-lösning för att förbättra tillgänglighet och noggrannhet av information om lokal badvattentemperatur på en ö i Stockholm skärgård. För att besvara syftet formuleras följande två frågeställningar:

1. Hur kan en IoT-lösning utformas för att effektivt samla in och överföra realtidsdata om badvattentemperaturen?
2. På vilket sätt kan informationen presenteras på ett användarvänligt sätt?

För att finna svar till projektets frågeställningar och därmed uppfylla syftet genomfördes en praktisk implementering av en sensorbaserad lösning. Denna lösning involverar dataöverföring via LoRa-nätverket till plattformen The Things Network som vidare integreras med Amazon Web Services för datalagring (Timestream-databas) och eventhantering (Lambda-funktion). Dessa respektive delar förser i sin tur en Grafana-dashboard och en Discord-server med data för visualisering.

Projektet resulterade i en heltäckande IoT-lösning för realtidsövervakning av vattentemperaturen. Tillgängligheten förbättrades genom att göra informationen lättåtkomlig via en webblänk. Noggrannheten förbättrades genom möjliggörandet av lokala temperaturmätningar. Dock återstår montering av hårdvaran vid en badbrygga för platsbaserade tester.

Avslutningsvis ges förslag på möjlig vidareutveckling av projektet särskilt avseende förbättrat hårdvaruskydd, jämförelsetal, egenutvecklad webbserver och robustare säkerhetslösningar.

Examensarbete

Summary

The project "An IoT solution for water temperature measurement" aims to develop an IoT solution to improve the availability and accuracy of information about local bathing water temperature on an island in the Stockholm archipelago. To answer the purpose, the following two questions are formulated:

1. How can an IoT solution be designed to efficiently collect and transmit real-time data about bath water temperature?
2. In what way can the information be presented in a user-friendly way?

In order to find answers to the project questions and thereby fulfill the purpose, a practical implementation of a sensor-based solution was carried out. This solution involves data transfer via the LoRa network to The Things Network platform which further integrates with Amazon Web Services for data storage (Timestream database) and event management (Lambda function). These respective parts in turn provides data to a Grafana dashboard and a Discord server for visualization.

The project resulted in a comprehensive IoT solution for real-time monitoring of the water temperature. Accessibility was improved by making the information easily accessible via a web link. Accuracy was improved by enabling local temperature measurements. However, the hardware remains to be installed at a bathing jetty for location-based tests.

In conclusion, suggestions are given for possible further development of the project, especially regarding improved hardware protection, comparison numbers, self-developed web server and more robust security solutions.

Innehållsförteckning

Figurförteckning

1. Inledning

Kapitlet introducerar problemområde och bakgrund som studien byggts upp kring. Vidare presenteras studiens syfte och frågeställningar. Kapitlet avslutas med studiens omfång och avgränsningar.

1.2 Bakgrund

Stockholms skärgård erbjuder under sommarhalvåret ett vatten med många attraktiva badplatser som lockar både besökare och de boende i skärgården[1]. Kännedom om vattentemperaturen är därför en viktig faktor för badsugna gäster, men kanske än mer för de boende i skärgården som har vattnet som en del av sin vardag. Allmänna badplatser kan ha platsbaserade termometrar men onlinebaserad information om lokal vattentemperatur är i många fall begränsat[2]. Visserligen finns temperaturen att tillgå hos havs- och vattenmyndigheten men då endast på ett antal utvalda platser[3].

Som delägare i ett sommarhus på en av öarna i skärgården har jag personligen upplevt denna brist på lokal temperaturinformation men det har även visat sig finnas ett behov bland ö:ns invånare att känna till. På den lokala Facebook-sidan har frågor kring vattentemperaturen ofta återkommit, vilket i dagsläget bara finns att läsa av vid badstegarna på respektive badplats. Idén till detta projekt växte därför fram ur en kombination av personlig erfarenhet och det uttryckta behovet bland öborna.

Figur 1 Mätstationer skärgården, Havs- och Vattenmyndigheten

[1] (Visit Stockholm)
[2] (Havs- och Vattenmyndigheten)
[3] (Havs- och Vattenmyndigheten)

1.3 Syfte

I bakgrunden framgår att information om lokal vattentemperatur är begränsad. Begränsningen är framförallt kopplad till online-baserad information, vilket medför osäkerhet vid bad. Därför är det relevant att skapa en egen lösning för att identifiera temperaturen och dela den med andra. Detta leder därmed vidare till studiens syfte:

Förbättra tillgängligheten och noggrannheten av information om lokal badvattentemperatur.

Genom att tillhandahålla vattentemperaturen på en online-baserad plattform, bidrar det till en ökad bekvämlighet för öns invånare och gör det möjligt för dem att fatta mer informerade beslut kring fritidsaktiviteter såsom bad.

1.4 Frågeställningar

För att kunna besvara syftet har det brutits ned i två övergripande frågeställningar. Dessa inriktar sig på den tekniska utformningen och genomförbarheten av en IoT-baserad lösning för temperaturövervakning och formulerades enligt följande:

1. Hur kan en IoT-lösning utformas för att effektivt samla in och överföra realtidsdata om badvattentemperaturen?
2. På vilket sätt kan informationen presenteras på ett användarvänligt sätt?

1.5 Avgränsningar

Projektet genomfördes med begränsade resurser i form av tid och budget, därför gjordes ett flertal avgränsningar i syfte att underlätta genomförbarheten. Delvis fokuserade projektet på att utveckla en lösning baserat på information från endast en sensor, det vill säga mätning från en enskild plats. Vidare koncentrerades datainsamlingen till vattentemperaturdata, utan insamling av ytterligare miljödata som exempelvis vattenkvalitet. För att spara ytterligare tid gjordes även datavisualiseringen grundläggande utan avancerade interaktiva funktioner eller anpassningar efter specifika användarpreferenser. Det genomfördes heller ingen integration med andra lokala system eller kommunikationstjänster utan projektet fokuserade på att utveckla en fristående lösning. Däremot integrerades två IoT-plattformar och visualiseringsprogram vilket ligger inom ramen för lösningen och beskrivs närmare i kapitel *2. Metodbeskrivning*. Viktigt att notera är att det även inte finns några planer för framtida uppgraderingar, dock resoneras kring vidare utveckling i kapitel *4. Diskussion*.

2. Metodbeskrivning

Kapitlet ger en översiktlig beskrivning av projektets utförande, design och metod för datainsamling och analys.

2.1 Implementeringsprocess

Projektet implementerades i tre huvudsakliga faser. Den inledande planeringsfasen bestod av en förstudie som syftade till att hitta ett problemområde. Denna fas inkluderade informationsinhämtning för att uppnå en grundförståelse och val av metod för datainsamling vilket resulterade i en projektplan som definierade projektets ramar (*Bilaga 1*). Fas 2,

genomförandefasen, inkluderade själva projektutförandet som innefattade inköp och sammankoppling av hårdvarukomponenter, insamlandet av telemetri, överföring och bearbetning av data samt testning och validering. Dessa steg beskrivs mer i detalj i *2.2 genomförande och testning*. Avslutningsvis, genomfördes sammanställningsfasen, vilket innefattade att skapa ett visualiseringslager och analysering av den insamlade telemetrin *(2.2.5 Effekt och datavisualisering*). Även här genomfördes tester och validering för att säkerställa kvaliteten på mätdata.

2.2 Genomförande och testning

Nedan beskrivs de specifika teknikerna som användes under projektets genomförande och testning. Här belyses praktiska aspekter från anslutning av hårdvara till systemkonstruktion som tillämpades för att säkerställa effektiv datainsamling och noggrannhet i resultaten.

2.2.1 Komponentbeskrivning

Den hårdvara som använts i projektet består av följande komponenter:

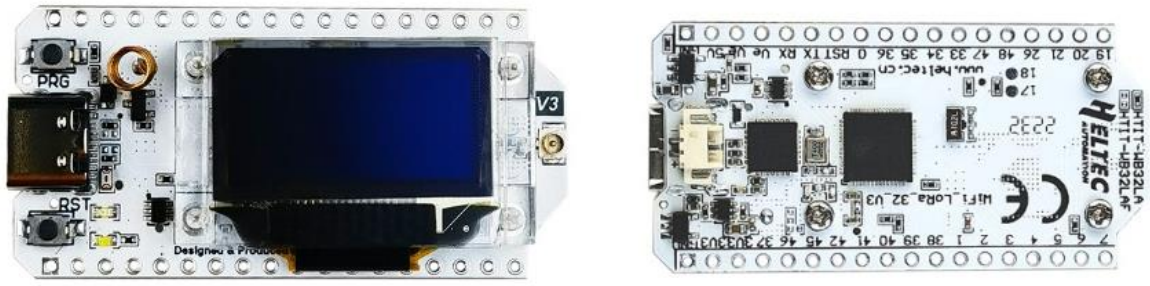
- DS18B20 temperatursensor
- Heltec WiFi LoRa 32 (ESP32-S3FN8) mikrokontroller
- Antenn för LoRa
- USB-C kabel
- Kablar hane/hona
- Li-Po-batteri 3,7 V

Vid inköp av hårdvara föll valet på DS18B20 temperatursensor. Denna sensor har en hög precision (±0,5 °C inom området -10 °C till +85 °C) och brett mätomfång (-55 °C till +125 °C) samt vattentäta egenskaper vilket gjorde att den lämpade sig väl i projektet[4]. Dessutom sitter sensorn längs med en längre kabel (91 cm) vilket ger lite distans för att komma ner i vattnet från exempelvis en brygga. Sensorn kommer med en adapter för att underlätta anslutning till en mikrokontroller på en strömförsörjning mellan 3-5 V, som även innehåller en inbyggd resistor (4,7 kΩ), för att stabilisera datalinjen och för att signalen ska kunna läsas korrekt av mikrokontrollern[5].

Figur 2 DS18B20 temperatursensor

[4] (Velleman, 2018)
[5] (Velleman, 2018)

Som mikrokontroller användes WiFi LoRa 32 (V3) från Heltec Automation. Denna mikrokontoller innehåller en ESP32-S3FN8 mikroprocessor, som med sin dubbelkärniga kapacitet på upp till 240 MHz, möjliggör effektiv databearbetning[6]. Den integrerar WiFi, LoRa (SX1262 LoRa-nodchip) och Bluetooth-anslutningar och har en 0,96-tums OLED-skärm för visning samt USB-C för programmering/strömförsörjning. Dessutom finns ett Inbyggt batterigränssnitt med ett integrerat litiumbatterihanteringssystem vilket säkerställer mobil energiförsörjning[7].



Figur 3 Wifi Lora 32 (V3)

Det medföljde en antenn för LoRa-kommunikation vid köpet av mikrokontrollern. Denna antenn hade dock dålig signalupptagningsförmåga, vilket beskrivs närmare i kapitel *4. Diskussion*. Under projektets arbete gick denna vid ett tillfälle sönder vilket gjorde att den byttes ut mot en antenn från Arduino med betydligt bättre upptagningsförmåga och som dessutom var vattentät[8].

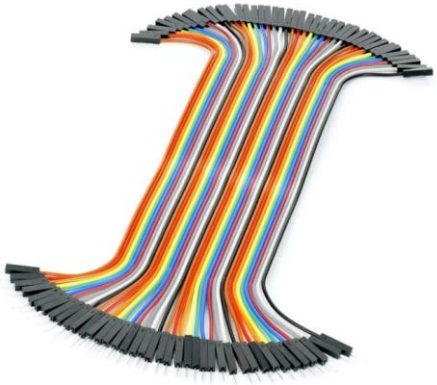
Figur 4 LoRa-antenner, från vänster: medföljande antenn till mikrokontrollern, Arduino antenn

USB-C kabel användes för att programmera mikrokontrollern och testa funktionaliteten. Under testfasen var det även denna kabel som strömförsörjde alla hårdvarukomponenter via datorn (USB-C till USB-C).

[6] (Heltec Automation)
[7] (Heltec Automation)
[8] (Arduino store)

Figur 5 USB-C kabel

För att sammankoppla sensorn med mikrokontrollern användes tre kopplingskablar av typen hane/hona. En för strömförsörjning (VCC), en för jord (GND) och en för output/dataledare som används för att kommunicera med mikrokontrollern.



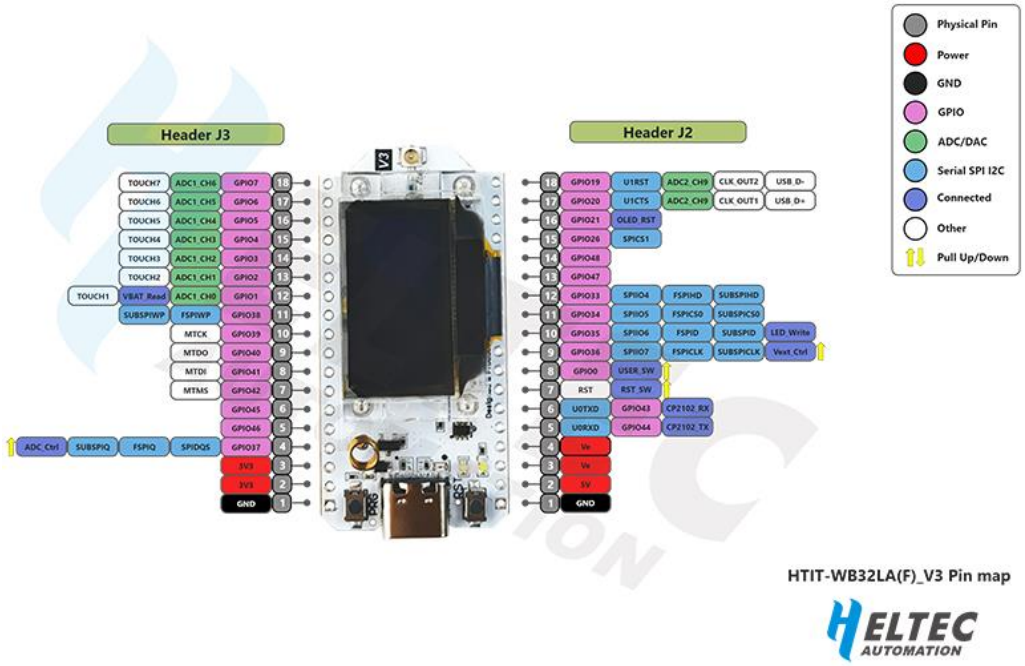
Figur 6 kopplingskablar hane/hona

Ett Litium Polymer (Li-Po) -batteri inhandlades i slutet av projektet när programkod för mikrokontrollen och lösningen var färdigt konfigurerad. Detta batteri gav en stabil energiförsörjning och skapade förutsättning att kunna montera hårdvaran på en brygga.

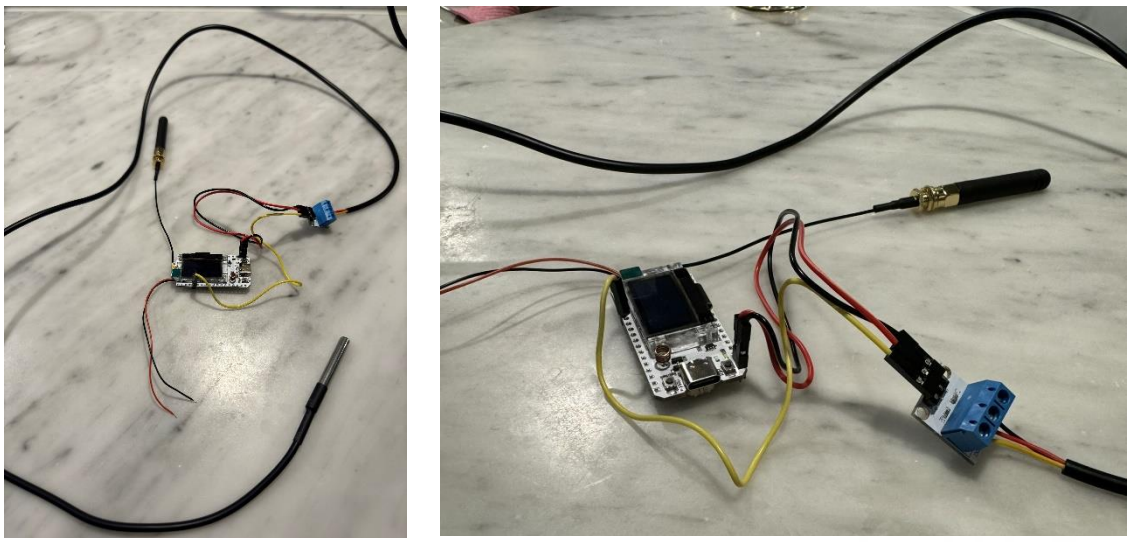
Figur 7 Li-Po-batteri

2.2.2 Sensorinstallation

Sammankoppling av hårdvara mellan sensorn och mikrokontrollern gjordes med utgångspunkt i enhetens pin-karta. Som strömförsöjning valdes 3.3V och dataledning (out) kopplades in på pin 3. Vidare kopplades jorden in på GND och den medföljande LoRa-antennen anslöts på avsedd plats.

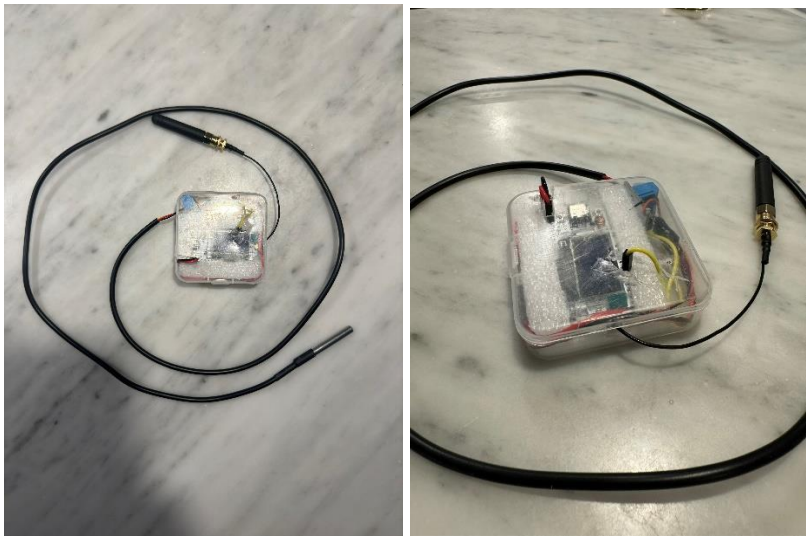


Figur 8 Pin-karta



Figur 9 Koppling sensor-mikrokontroller

Mikrokontrollern levererades i en liten plastlåda. Denna låda anpassades för att kunna hålla ihop alla hårdvarukomponenter och formade även ett skyddande hölje.



Figur 10 Hölje

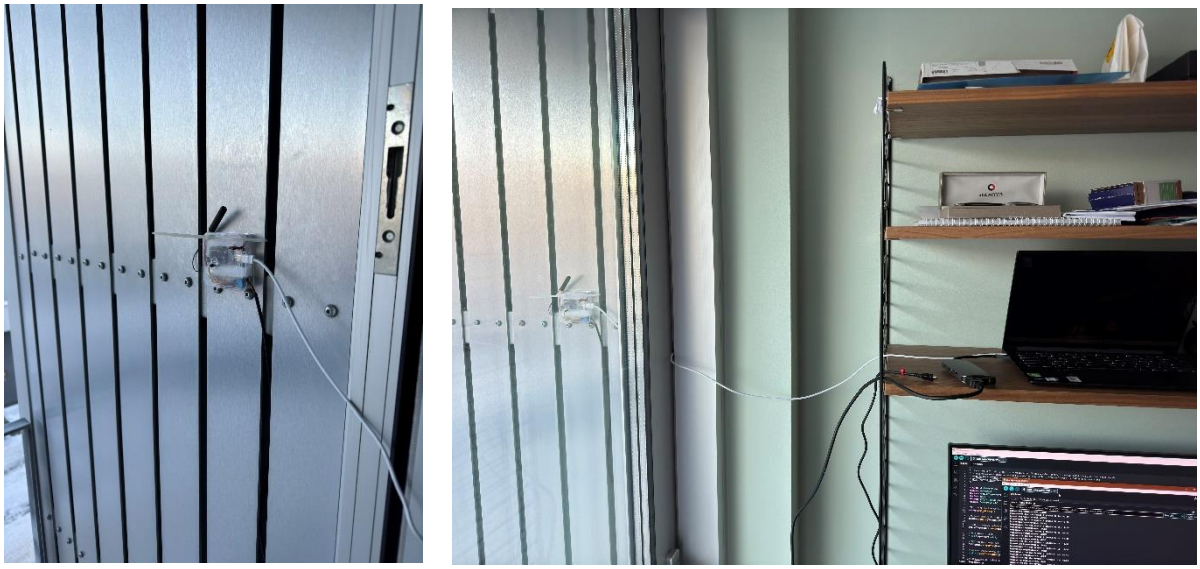
Eftersom LoRa användes för nätverkskommunikation var enheten tvungen att placeras utomhus för att få täckning och tillräcklig signalstyrka för dataöverföring. Detta gjordes genom att fästa enheten på en balkongvägg med hjälp av buntband. Sedan kopplades USB-C kabeln till datorn för strömförsörjning och programmering av enheten. Programmeringen gjordes i Arduino IDE och baserades på de officiella biblioteken från ”Heltec esp32+LoRa” samt ”Dallas Temperature” för sensorn[9][10]. För fullständig kod se Github[11]. Hur programmet är uppsatt beskrivs närmare i kapitel *2.2.4 Plattform och datahantering*.

main.ino credentials.h

```
1  /*
2  * LoRaWan and OLED code is based on examples from official Arduino library for Heltec ESP32 (or ESP32+LoRa) based boards
3  * DS18B20 temperature sensor code based on examples from official Arduino library for Dallas Temperature ICs
4  * https://github.com/Heltec-Aaron-Lee/WiFi_Kit_series
5  * https://github.com/milesburton/Arduino-Temperature-Control-Library
6  */
7
8  #include <LoRaWan_APP.h>
9  #include <OneWire.h>
10 #include <DallasTemperature.h>
11 #include <Wire.h>
12 #include <HT_SSD1306Wire.h>
13 #include "credentials.h"
```

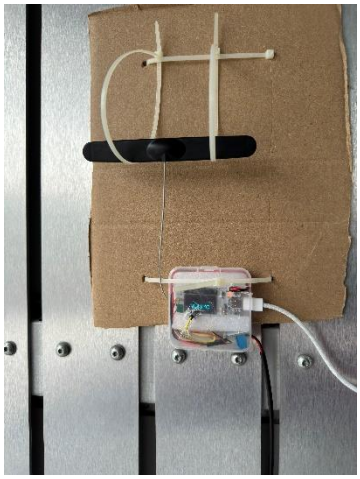
Figur 11 Programbibliotek

[9] (Github Heltec library)
[10] (Github temperature library)
[11] (Github project repository)



Figur 12 Uppsättning balkongvägg

Som tidigare nämnt i kapitel *2.2.1 Komponentbeskrivning*, gick antennen vid ett tillfälle på grund av en olycka sönder. Detta innebar att en ny antenn fick inhandlas och monteras upp. Detta var dock tur i oturen då den tidigare antennen försedde enheten med en mycket svag signal vilket gjorde det svårt att komma igång med lösningen, mer om det i kapitel *Diskussion*. Den nya antennen hängdes upp framför ett underlag av kartong, eftersom metall inte var rekommenderat enl. tillverkarens dokumentation[12].



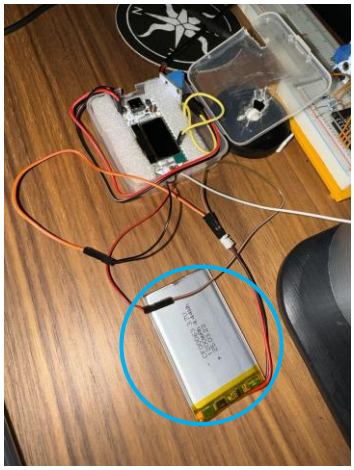
Figur 13 Installering av Arduino antenn

[12] (Arduino store)

Sensorn placerades slutligen i ett glas med vatten för att simulera badvattnet. Notera dock att större del av testningen har dock gjorts baserat på utomhusluften då det rådde -10 grader i utomhustemperatur under testperioden så allt vatten frös. Men eftersom sensorn fungerar lika bra i luft som i vatten så har det ingen betydelse för det slutliga användningsområdet[13].

Figur 14 Sensor i vatten

Den sista delen av hårdvaruinstallationen var att koppla in en extern strömkälla i form av ett batteri för att sedan kunna använda sensorn vid en badbrygga. Detta gjordes som ett sista steg i implementeringsprocessen när all kod var konfigurerad och lösningen var uppsatt och klar.

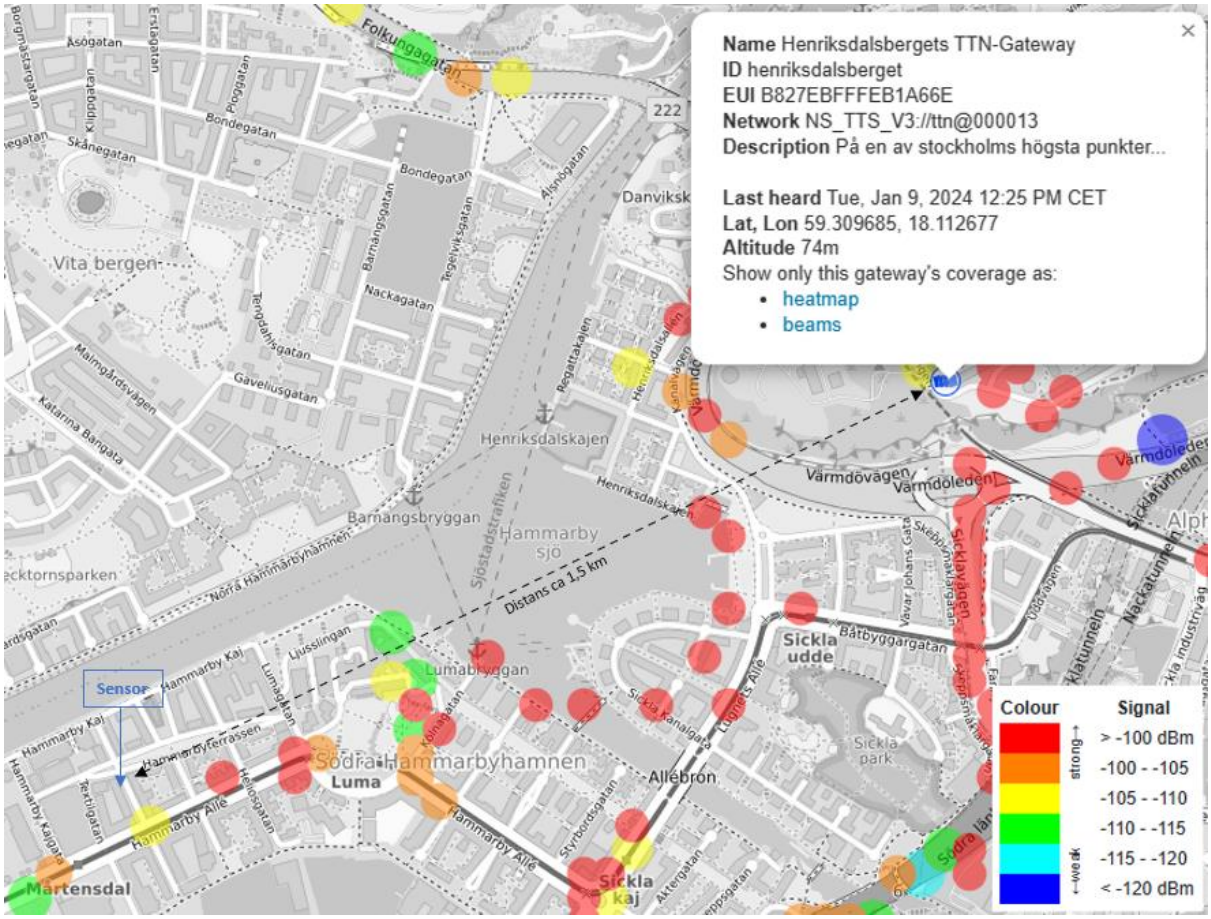


Figur 15 Batteriinstallation

[13] (Velleman, 2018)

2.2.3 Nätverk och dataöverföring

I detta projekt användes LoRaWan-nätverket. LoRa står för ”Long Range” och är teknik för trådlös dataöverföring som möjliggör överföring av data över långa sträckor, ofta flera kilometer, med låg energiförbrukning. Jämfört med tekniker som WiFi, Bluetooth eller ZigBee går det att kommunicera på betydligt längre avstånd[14]. Detta gör det väl lämpat för detta IoT-projekt där små mängder temperaturdata ska skickas från en badbrygga dit WiFi inte når[15]. LoRa använder licensfria frekvenser på UHF (Ultra high Frequency) och VHF-banden (Very High Frequency), det vill säga på frekvenser som ligger under 1 GHz[16]. För detta projekt har frekvensen 868 MHz (Europa) använts. Konfiguration och testning av lösningen genomfördes med en distans på ca 1,5 kilometer till närmsta gateway enl. *Figur 16 Avstånd till gateway.*

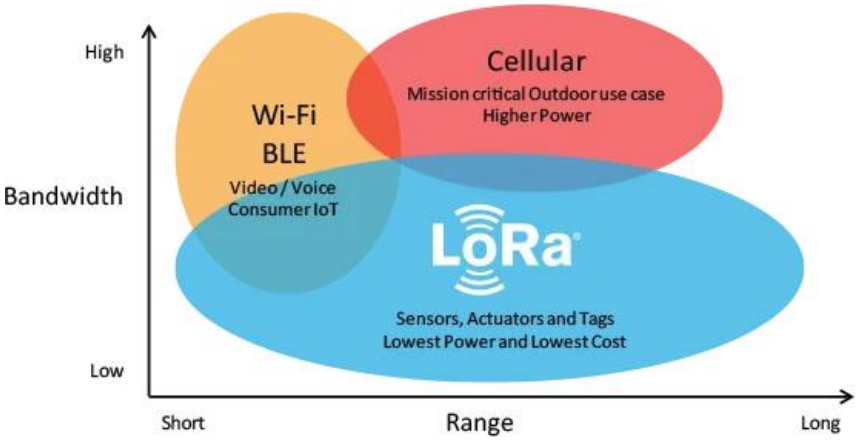


Figur 16 Avstånd till gateway

LoRaWan används i dagligt tal ibland synonymt med LoRa men är själva nätverksprotokollet (Low Power Wide Area Networks, LPWAN) som hanterar kommunikationen mellan gateways

[14] (The Things Network)
[15] (The Things Network)
[16] (Induo)

och noder (IoT-enheter)[17]. Gateways länkar i sin tur ihop noderna med det vanliga vanliga IP-nätverket, med andra ord en brygga som omvandlar inkommande radiopaket till IP-paket. Ett datapaket som i denna IoT-lösning gick vidare till en plattform kallad The Things Network (TTN) vilket beskrivs vidare i kapitel *2.2.4 Plattform och datahantering*.



Figur 17 LoRa vs WiFi etc.

2.2.4 Plattform och datahantering

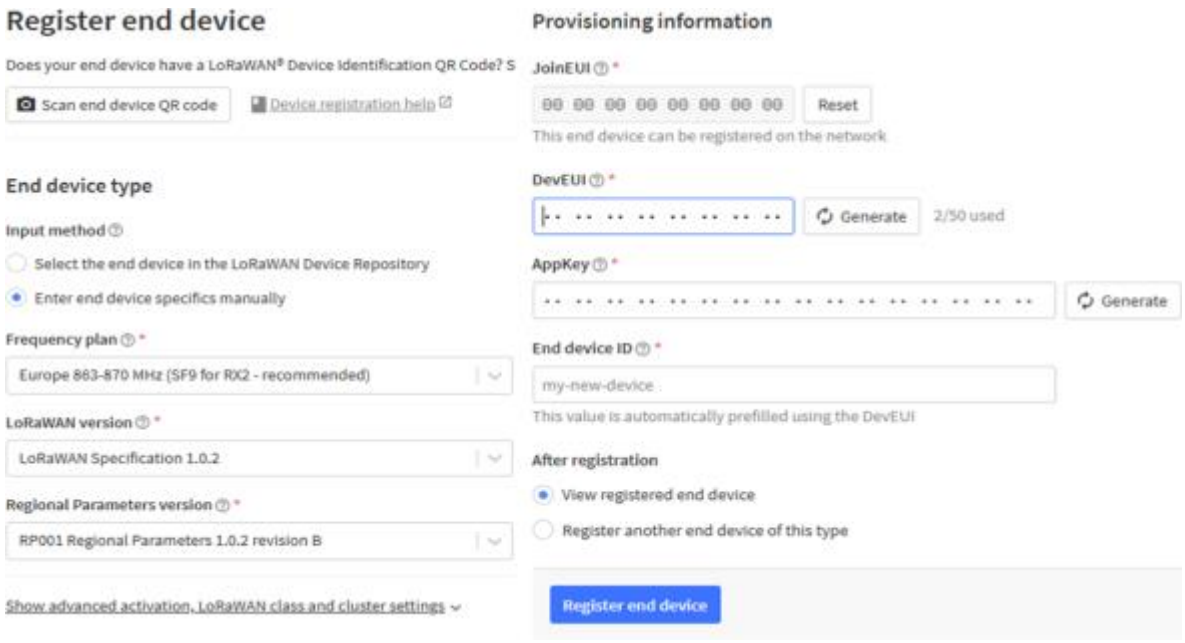
Två huvudsakliga plattformar användes i lösningen: The Things Network (TTN) och Amazon Web Service (AWS). För att hantera inkommande datapaket från enheten via LoRa-nätverket användes TTN. TTN är en global community som syftar till att bygga upp ett decentraliserat och öppet nätverk för IoT genom LoRaWan[18]. De har en plattform de kallar ”The Things Stack”, en LoRaWan-nätverksserver som hanterar kommunikationen från enheter på nätverket[19]. ”The Things Stack” erbjuder fyra olika typer av plattformar:

- The Things Stack Cloud
- The Things Stack Enterprise
- The Things Stack Sandbox
- The Things Stack Open Source

I detta projekt har den The Things Stack Sandbox använts för att sätta upp lösningen. Konfigurationen inleddes med att skapa en applikation på plattformen med ett tillhörande id. Sedan registrerades en ”End device”. För att registrera en enhet behövs ett antal parametrar specificeras som frekvensområde, LoRaWan-version och ett antal nycklar för identifiering.

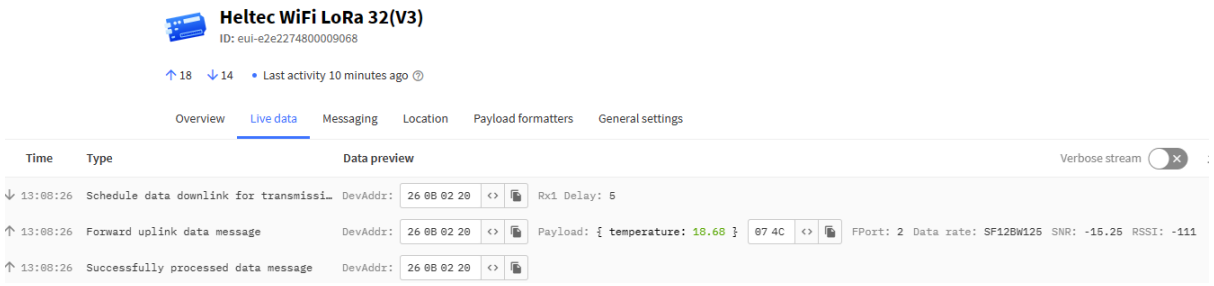
[17] (Induo)
[18] (The Things Network)
[19] (The Things Stack)

Dessa nycklar hämtades delvis från enhetstillverkarens dokumentation[20] men genererades även i TTN och lades i en separat Header-fil i koden[21].



Figur 18 Enhetsregistrering

När anslutningen sedan var fullbordad gick det att följa den data som kom in till TTN via ”Live data”. Den inkommna data kom dock in på bitformat, så för att kunna se själva temperaturvärdet behövdes en såkallad ”payload formatter” (JavaScript) skapas, vilket omvandlade datapaketet från bitnivå till läsbar temperatur.



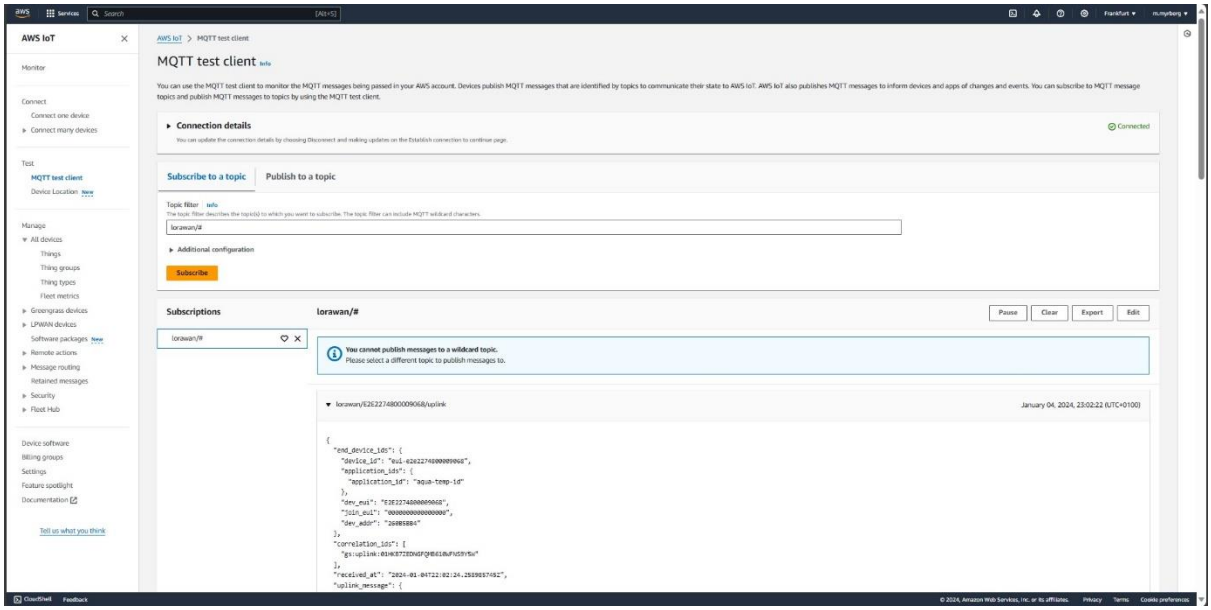
Figur 19 Live data TTN

[20] (Heltec documentation)
[21] (Github project repository)

Figur 20 Payload formatter

När kommunikationen med TTN var uppsatt och stabil, gick arbetet arbetet med lösningen över till att integrera AWS. Denna integration genomfördes via TTN:s ”Deployment Guide” som är ett smidigt sätt att integrera AWS med hjälp av en AWS CloudFormation-mall[22]. Därefter kan man börja följa sin data i MQTT-test-klienten på AWS IoT Core.

[22] (Deployment Guide)



Figur 21 Mqtt test client

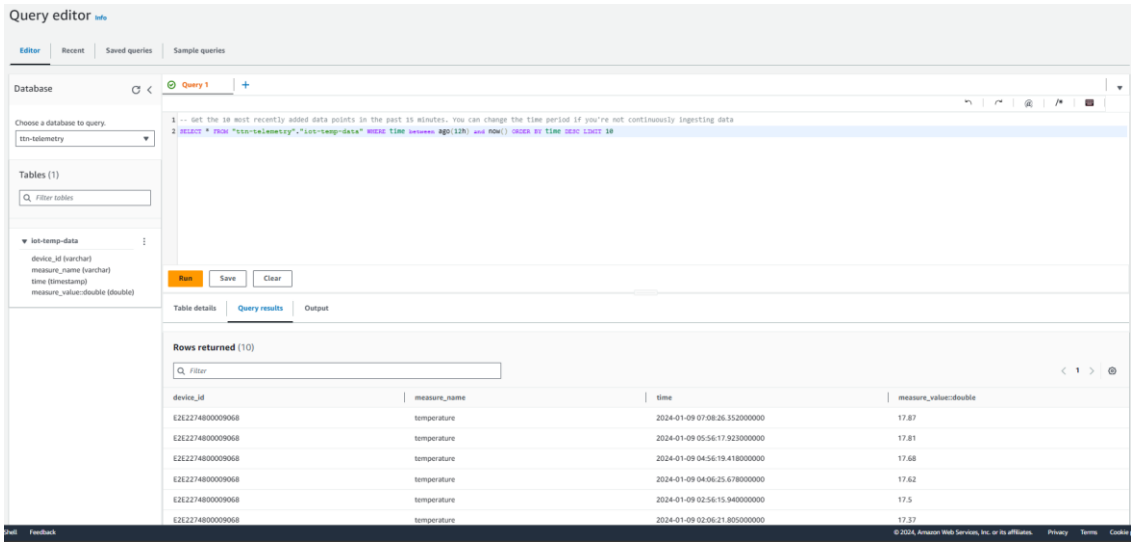
Sedan konfiguerades “Message Routing-regler”. Med dessa regler går det att via bearbeta inkommande meddelanden från TTN och vidarebefordra dem till ett AWS-verktyg[23]. I lösningen bestod dessa ”verktyg” av en Timestream-databas och en Lambda-funktion.

- Timestream-regeln: Alla inkommande meddelanden till IoT Core dirigeras till Timestream-databasen genom regeln (SQL-syntax):

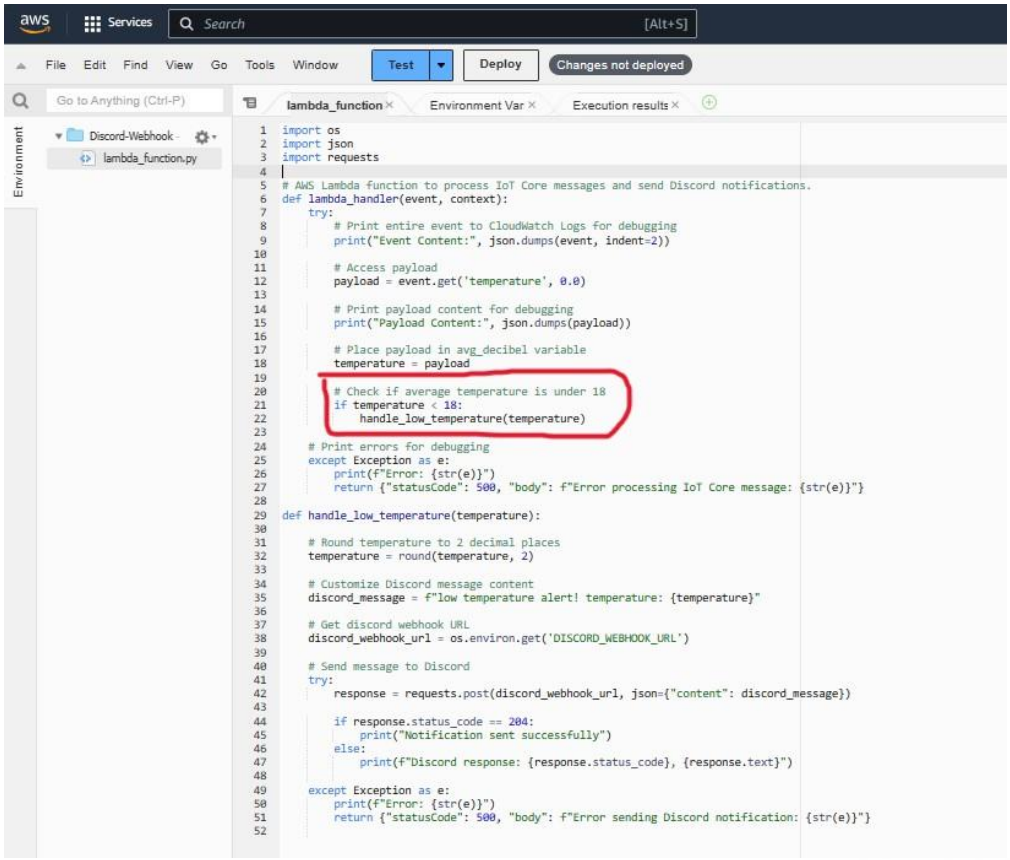
$$SELECT\ uplink\_message.decoded\_payload.temperature\ as\ temperature\ FROM\ 'lorawan/+/uplink'$$

- Lambda-regeln: Tar emot meddelanden från IoT Core. För alla temperaturvärden som understiger 18 grader, initierar en Lambda funktion (Python-script) en aviseringsåtgärd till Discord via ett Webhook API. Discord-kanalen tar emot meddelandet och ger en avisering till alla anslutna medlemmar på discord-servern.

[23] (AWS Documentation)



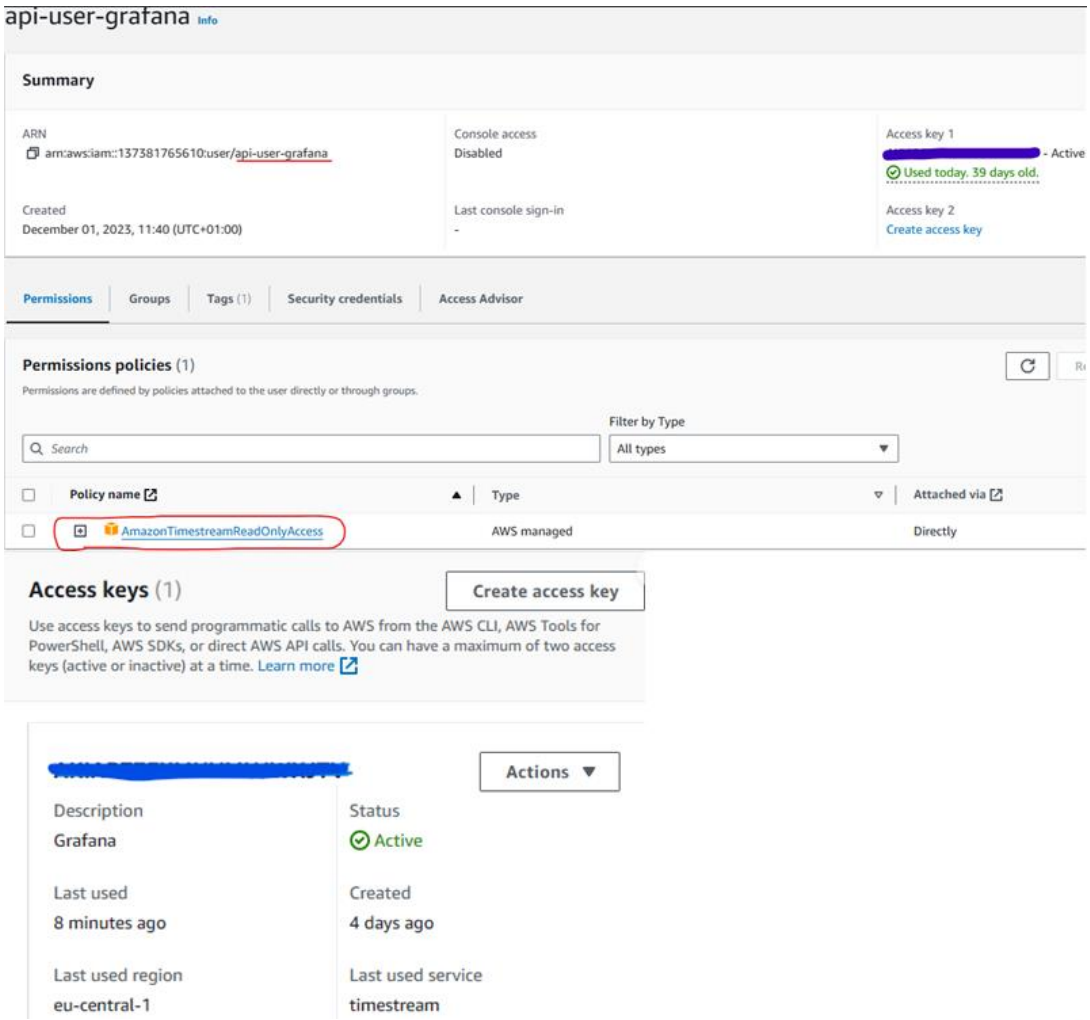
Figur 22 Timestream-databas



Figur 23 Lambda funktion

2.2.5 Effekt och datavisualisering

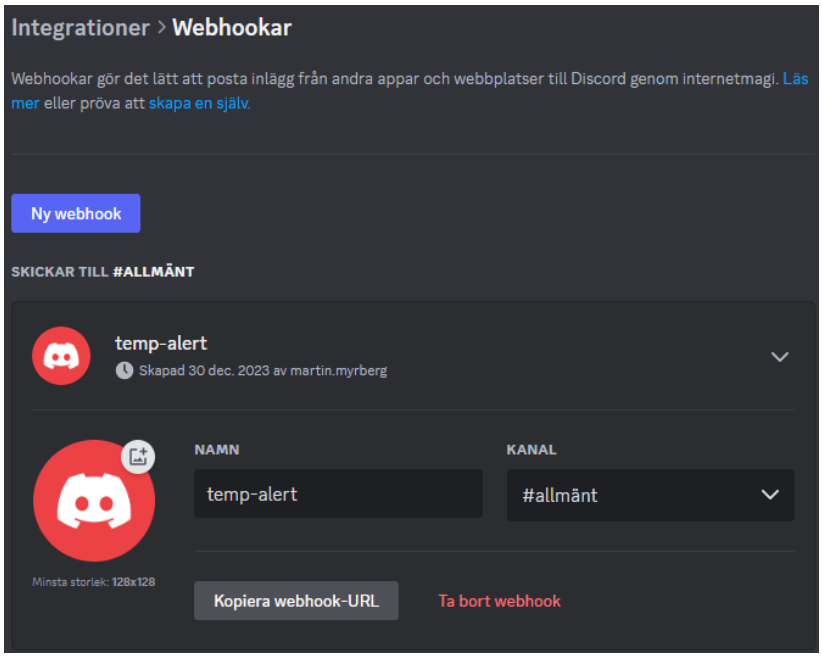
Den avslutande delen av denna IoT-lösning innefattade att bygga det så kallade effektlagret eller användargränssnittet. Här skapades en dashboard i Grafana samt notisfunktion till Discord vilka visas i kapitel *3. Resultat*. För att lyckas integrera AWS med Grafana och implementera datavisualiseringen skapades en ”IAM-user” (Identity and Access Management) med namnet ”api-user-grafana ” med tillhörande åtkomstnycklar. IAM är en tjänst i AWS som används för att hantera åtkomst till olika AWS-tjänster och resurser[24]. Grafana ställdes sedan in för att använda dessa nycklar för interaktion med Timestream-databasen. Dessutom applicerades IAM-policyer på api-user-grafana, specifikt ”AmazonTimestreamReadOnlyAccess”, vilket ger definierade behörigheter för Grafana vid interaktion med Timestream-databasen.



Figur 24 IAM roll och nyckel

[24] (AWS Documentation)

Som tidigare beskrivet implementerades även en Lambda-funktion (*Figur 23 Lambda funktion*) i AWS för att trigga en notifikation till en Discord-server via ett webhook-api. Discord har en integrationsdel där en URL för webhook skapades och implementerades i Lambda-funktionen.



Figur 25 Discordintegration

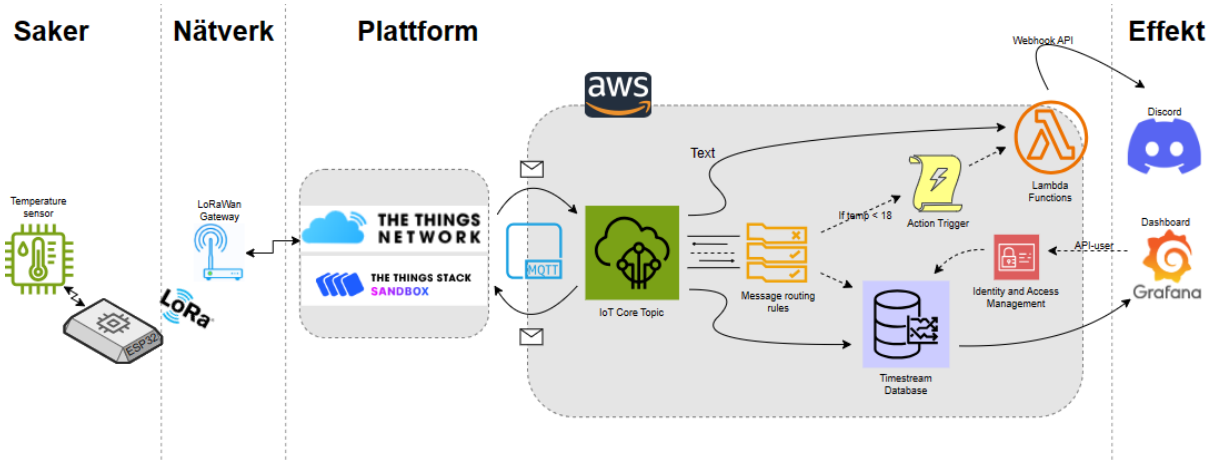
3. Resultat

Kapitlet ger en sammanfattande beskrivning av studiens resultat utifrån tidigare metodbeskrivning.

Projektet baserar sina resultat på frågeställning 1 och den genomförda metodbeskrivningen. Dessa resultat tillämpas sedan för att besvara frågeställning 2 och skapa en heltäckande IoT-lösning för badvattentemperatur.

3.1 Hur kan en IoT-lösning utformas för att effektivt samla in och överföra realtidsdata om badvattentemperaturen?

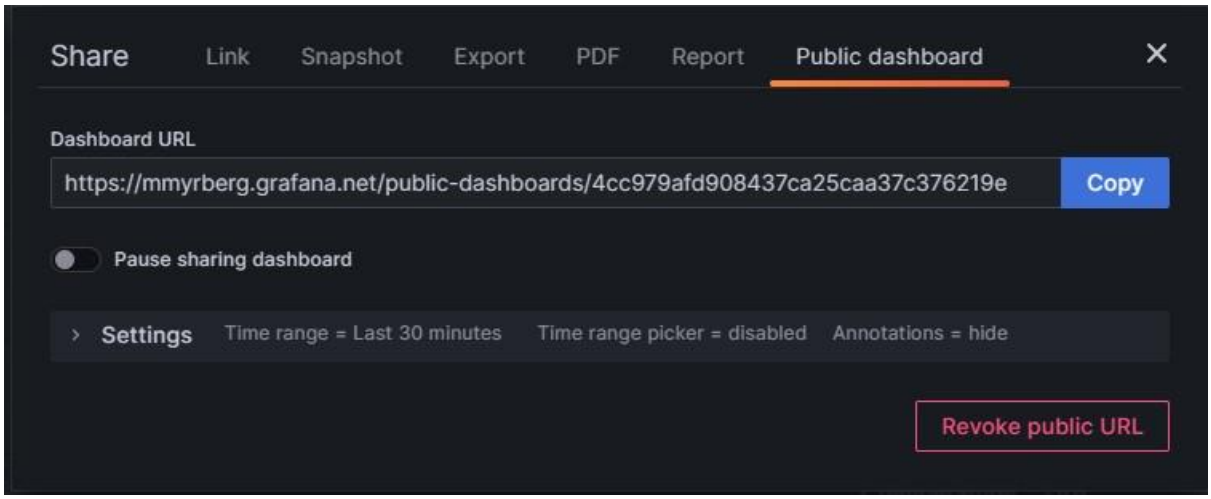
Projektet resulterade i följande lösningsarkitektur enligt *Figur 26 IoT-arkitektur*. Detta beskriver den övergripande utformningen av IoT-lösningen och besvarar frågeställningen. Notera att intervall för realtidsdata (datainsamling) i nuvarande lösning ligger på 1ggr/h. *Figur 26 IoT-arkitektur* visas i förstorad upplaga i *bilaga 2*.



Figur 26 IoT-arkitektur

3.2 På vilket sätt kan informationen presenteras på ett användarvänligt sätt?

Baserat på frågeställning 1 har en medvetenhet om hur information kan presenteras på ett användarvänligt sätt erhållits. Svaret på denna fråga resulterade därför i ett användargränssnitt med en tydlig Grafana-dashboard. Denna gjordes publik vilket innebär att alla med länk kan komma åt sidan via en webbläsare. Vidare gjordes visualiseringen så att temperaturen skiftar till en blå (kall) färg om badvattentemperaturen understiger 18 grader i syfte att öka infografiken och underlätta förståelsen.



Figur 27 Länk till dashboard

Figur 28 Grafana dashboard

Slutligen gav uppsättningen av ett webhook-api mot en Discord-server en notifieringsfunktion som resulterade i automatiserad informationsåtkomst och användarvänlig kommunikation.

Figur 29 Discord notifikation

4. Diskussion

Kapitlet diskuterar projektets resultat, implikationer och begränsningar samt ger förslag på vidare utveckling.

Projektet har uppfyllt projektplanens mål med att skapa en heltäckande IoT-lösning från sensor till effekt i form av en digital plattform (dashboard) för att visa badvattentemperaturen i realtid. En initial osäkerhetsfaktor i projektplanen gällde integrationen av ytterligare en plattform. Detta visade sig vara smidigare än förväntat tack vare TTN:s effektiva integrationsfunktionalitet med AWS. Däremot uppstod tekniska svårigheter när det gällde

anslutningen till TTN via LoRa-nätverket. Problemet uppstod till följd av en antenn med otillräcklig signalupptagningsförmåga, och med mina begränsade erfarenheter av LoRa var denna felsökning svår och tog mycket tid. Detta löste sig dock oväntat när den ursprungliga antennen gick sönder och ersattes av en ny som fungerade betydligt bättre, men detta hade antagligen inte upptäckts om den inte gått sönder. Det finns en förväntan att tillverkaren ska leverera enligt deras specifikation, vilket belyser vikten av att noggrant granska och validera leverantörer före inköp till praktiska tillämpningar.

När det gäller projektets syfte, har lösningen möjliggjort både förbättrad tillgänglighet och noggrannhet i mätningarna av badvattentemperaturen. Tillgängligheten förbättrades genom att göra informationen om badvattentemperaturen lättåtkomlig via en webblänk, vilket står i kontrast till den mer tidskrävande processen att hämta informationen från Havs- och vattenmyndighetens webbplats. Samtidigt uppdaterar de endast temperaturen en gång per dygn[25], till skillnad från denna lösning som uppdaterar varje timme. När det gäller noggrannheten, har projektets lösning möjliggjort lokala mätningar av vattentemperaturen. Detta är särskilt relevant då temperaturen kan variera mellan olika badplatser och därmed skilja sig från de platser Havs- och vattenmyndigheten utför sina mätningar.

Projektet har utvecklats under vintern med frusna sjöar och vatten vilket begränsat mina möjligheter att testa lösningen i en faktisk användarmiljö. Det återstår alltså att montera sensorn vid en specifik badbrygga. Denna fas är viktig eftersom det först är när utrustningen är helt platsmonterad som det är möjligt att grundligt utvärdera och bekräfta systemets mätnoggrannhet under verkliga förhållanden. Här behövs vidare utveckling av det fysiska skyddet/höljet för hårdvaran och dess komponenter för att det ska klara de stundvis tuffa väderförhållandena vid en badbrygga. Om systemet visar sig fungera väl över en utvärderingsperiod, är det relevant att titta på att installera fler sensorer för att täcka in flera badplatser i lösningen.

Som ytterligare vidareutveckling skulle det även vara intressant att skapa funktionalitet för att jämföra temperaturen med den offentliga mätningen från havs- och vattenmyndigheten. Detta skulle kunna genomföras med deras öppna data/api[26] för att hämta in och lagra värdena i Timestream-databasen och sedan visualisera jämförelsen i Grafana. Därigenom har användaren något att sätta mätvärdena i relation till, vilket ökar relevansen i lösningen.

En nackdel med denna lösning är att den kostar att underhålla. I detta fall är det AWS som tar betalt för sina tjänster, framförallt Timestream-databasen[27]. Som vidareutveckling skulle det vara intressant att titta på att utveckla en egen webbserver med en lokal databas och en webbsida för datavisualisering. Ett sådant steg kan minska de ekonomiska kostnaderna. Det bör dock noteras att detta alternativ kan kräva mer omfattande underhåll och innebära högre tidsåtgång under själva utvecklingsfasen.

[25] (Havs- och vattenmyndigheten)
[26] (Havs- och vattenmyndigheten)
[27] (AWS pricing)

Slutligen är säkerhet en annan viktig aspekt som kan vidareutvecklas i detta projekt. Nuvarande lösning har implementerats med de olika plattformarnas standardinställningar för säkerhet i syfte att snabbt få lösningen fungerande. Här finns det möjlighet att forska vidare och säkerställa att sensordata hanteras och överförs med optimal säkerhet för att undvika obehörig åtkomst eller manipulation. Det skulle exempelvis kunna handla om allt från förstärkt kryptering till att implementera mer sofistikerade autentiseringsmekanismer eller att integrera tredjepartslösningar som är specialiserade på just IoT-säkerhet.

5. Slutsatser

Kapitlet beskriver projektets slutsatser.

Utifrån resultatet går det att dra slutsatsen att den utvecklade IoT-lösningen framgångsrikt tillhandahåller information om badvattentemperaturen. Genom att tillämpa systemarkitekturen som framgår i *Figur 26 IoT-arkitektur*, uppnås en hög grad av både noggrannhet i mätningarna och tillgänglighet för användarna. Även om projektet påvisar önskvärt resultat belyses också betydande möjligheter för vidare utveckling av lösningen. Särskilt avseende förbättrat hårdvaruskydd, jämförelsetal, egenutvecklad webbserver och robustare säkerhetslösningar. Förhoppningsvis kan detta projekt inspirera till nya innovativa lösningar och driva på fortsatt utveckling inom IoT.

6. Referenser

Arduino store. (u.d.). Hämtat från "Dipole Pentaband Waterproof Antenna": https://store.arduino.cc/products/dipole-pentaband-waterproof-antenna?selectedStore=eu

AWS Documentation. (u.d.). Hämtat från "What is IAM?": https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html

AWS Documentation. (u.d.). Hämtat från "Creating AWS IoT rules to route device data to other services": https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules-tutorial.html

AWS pricing. (u.d.). Hämtat från "Amazon Timestream pricing": https://aws.amazon.com/timestream/pricing/

Deployment Guide. (u.d.). Hämtat från "Learn how to deploy the AWS IoT integration for The Things Stack": https://www.thethingsindustries.com/docs/integrations/cloud-integrations/aws-iot/deployment-guide/

Github Heltec library. (u.d.). Hämtat från "Heltec ESP32 & ESP8266 Series Arduino Develop Environment": https://github.com/Heltec-Aaron-Lee/WiFi_Kit_series/tree/master

Github project repository. (u.d.). Hämtat från "AquaTherm-monitoring-system": https://github.com/mmyrberg/AquaTherm-monitoring-system

Github temperature library. (u.d.). Hämtat från "Arduino Library for Maxim Temperature Integrated Circuits": https://github.com/milesburton/Arduino-Temperature-Control-Library

Havs- och vattenmyndigheten. (u.d.). Hämtat från "Öppna data och statistik": https://www.havochvatten.se/data-kartor-och-rapporter/data-och-statistik.html

Havs- och vattenmyndigheten. (u.d.). Hämtat från "Ekhagens strandbad": https://www.havochvatten.se/badplatser-och-badvatten/kommuner/badplatser-i-stockholms-kommun/ekhagens-strandbad.html

Havs- och Vattenmyndigheten. (u.d.). Hämtat från "Vattentemperatur och kvalitet på badvatten": https://www.havochvatten.se/badplatser-och-badvatten/kommuner/badplatser-i-stockholms-kommun.html

Heltec Automation. (u.d.). Hämtat från "WiFi LoRa 32(V3)": https://heltec.org/project/wifi-lora-32-v3/

Heltec documentation. (u.d.). Hämtat från “ESP32+LoRa Preparation & Config Parameters": https://docs.heltec.org/en/node/esp32/esp32_general_docs/lorawan/index.html

Induo. (u.d.). Hämtat från "VAD ÄR LORA? OCH VAD MENAS MED LORAWAN?": https://www.induo.com/s/g/iot-internet-of-things/vad-ar-lora/

The Things Network. (u.d.). Hämtat från "What are LoRa and LoRaWAN?": https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/

The Things Network. (u.d.). Hämtat från "Homepage": https://www.thethingsnetwork.org/

The Things Stack. (u.d.). Hämtat från "The Things Stack Documentation": https://www.thethingsindustries.com/docs

Velleman. (den 13 03 2018). Hämtat från "TEMPERATURE PROBE DS18B20 & ARDUINO® COMPATIBLE ADAPTER": https://cdn.velleman.eu/downloads/29/vma324_a4v01.pdf

Visit Stockholm. (u.d.). Hämtat från "Upptäck Stockholms skärgård": https://www.visitstockholm.se/se-gora/utflykter/upptack-stockholms-skargard/

7. Bilagor

Bilaga 1

PROJEKTPLAN EXAMENSARBETE

PROJEKTNAMN: Badvattentemperatur (GitHub: "AquaTherm-monitoring-system")
UTFÖRARE: Martin Myrberg (inga fler involverade personer)
KLASS: IOT22
DATUM: 2023-12-15

PROBLEMFORMULERING

Jag har landställe på en ö i Stockholms skärgård där det finns ett par badplatser tillgängliga för allmänheten. Lokal vattentemperatur är något som boende på ön ofta vill veta men som i dagsläget bara finns att läsa av vid badstegarna på respektive badplats. På ö:ns lokala Facebook-grupp återkommer frågan: ”Någon som varit nere och badat och kan berätta hur många grader det är?”.

SYFTE/MÅL, VAD SKA PROJEKTET UPPNÅ?

Syfte:

- Förbättra tillgängligheten och noggrannheten av information om badvattentemperaturen för boende på ön. Genom att tillhandahålla vattentemperaturen på en online-baserad plattform, bidrar det till en ökad bekvämlighet för öns invånare och gör det möjligt för dem att fatta mer informerade beslut kring fritidsaktiviteter såsom bad.

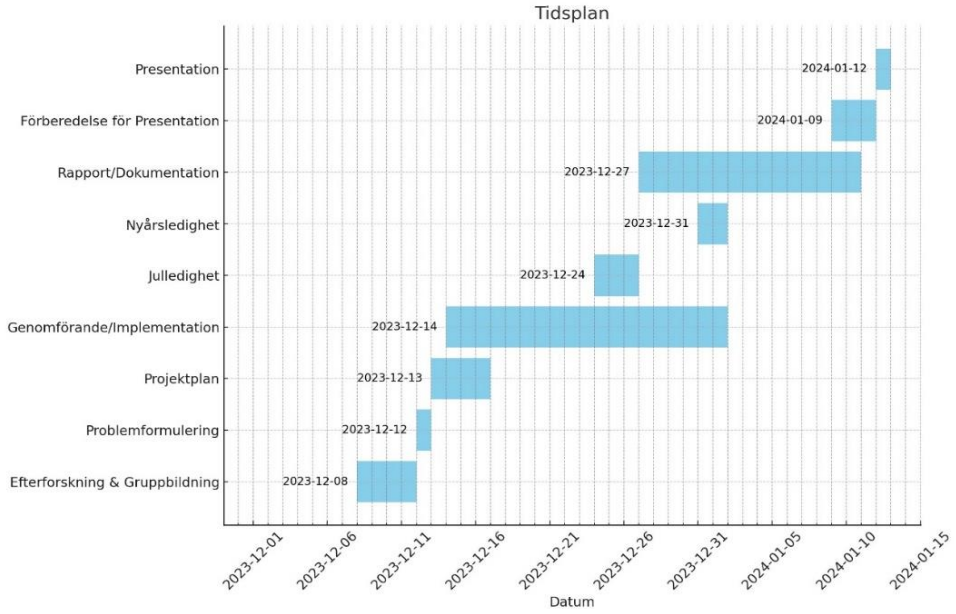
Mål:

- Att utveckla och implementera en användarvänlig IoT-lösning som samlar in, överför och presenterar realtidsdata om badvattentemperaturen.

TIDPLAN, NÄR SKA PROJEKTET UPPNÅS?

Detta projekt skall levereras inom en specificerad tidsram, projektets hålltider är enligt följande:

- **Research val av projekt och gruppbildning:** 2023-12-08 – 2023-12-11
- **Problemformulering:** 2023-12-12
- **Projektplan:** 2023-12-13 – 2023-12-16
- **Genomförande/Implementation:** 2023-12-14 – 2024-01-01 (Obs! inget arbete under jul- och nyårshelgerna)
- **Rapport/Dokumentation:** 2023-12-27 – 2024-01-10
- **Förberedelse för Presentation:** 2024-01-09 – 2024-01-11
- **Presentation:** 2024-01-12



METOD, HUR SKA PROJEKTET UPPNÅS?

Metodavsnittet ska ses som en initial plan/intention för hur projektet syfte och mål ska realiseras. Metodiken som sedan faktiskt implementerades i projektarbetets utförande beskrivs i slutrapporten.

Datainsamling:

- Sensorval: Användning av DS18B20 temperatursensor för mätning av vattentemperaturen. Har hög precision och vattentäta egenskaper. Sensorinstallationen baseras på rekommendationer från officiella källor som datablad och produktbeskrivningar.
- Dataöverföring: Implementering av *Heltec WiFi LoRa 32 (ESP32-S3FN8) mikrokontroller* för sensorkommunikation och dataöverföring till The Things Network (TTN) via LoRaWAN. LoRa-tekniken används på grund av dess långa räckvidd som behövs för nätverksåtkomsten vid denna badplats. Mikrokontroller + sensor planeras att drivas av AA-litiumbatterier som fungerar i extrema temperaturförhållanden (-30 till 60 °C).
- Insamlingsfrekvens: Fastställande av intervall för datainsamling kommer baseras på miljöförhållanden och användarbehov. Utgångspunkt 1ggr/h.

Datahantering och analys:

- Molntjänst: Användning av TTN för datahantering, lagring och visualisering om möjligt. Information om integrationer och säkerhetsåtgärder inhämtas från The Things Networks officiella dokumentation samt användarforum och github för kodrelaterade sektioner.
- Integration: Möjligtvis integrera TTN med AWS för att kunna skapa en lambdafunktion som implementerar ett Discord webhook-API för notifikationer samt eventuellt en Timestream databas som förser en Grafana dashboard med data. Beror på hur mycket som är möjligt att göra direkt i TTN.

Teknisk Efterforskning och Utveckling:

- Primärkällor: Användning av officiella källor som datablad och tekniska specifikationer för information om hårdvarukomponenter och plattformen.
- Sekundärkällor: Användning av artiklar/bloggar, forum som Stackoverflow, YT-videor och Github för ytterligare kunskap, särskilt kring kodutveckling och systemintegration.

- Experthjälp: Söka stöd från lärare och studenkollegor för att lösa tekniska och projektrelaterade frågor när det är behövligt.

Användning av AI:

- Komplementär Hjälp: Användning av AI-tjänster som ChatGPT för att pedagogiskt förklara och hjälpa till med oklara delar av projektet. Komplement till primära och sekundära källor.

AVGRÄNSNINGAR

Geografi:

- Projektet är avgränsat till att mäta temperaturen på en specifik badplats på ön. Detta innebär att ingen utvidgning till andra badplatser planeras inom ramen för detta projekt.

Datainsamling:

- Endast vattentemperaturdata samlas in. Projektet kommer inte att innehålla insamling av ytterligare miljödata såsom vattenkvalitet eller biologiska parametrar.

Användargränssnitt och Interaktivitet:

- Användargränssnittet för att visa data kommer att vara grundläggande och fokuserat på att visa vattentemperatur. Det kommer inte att innefatta avancerade interaktiva funktioner eller anpassning baserat på individuella användarpreferenser.

Integration med andra system:

- Projektet kommer inte att integrera med andra lokala informationssystem, turistapplikationer eller kommunikationstjänster. Fokus ligger enbart på att utveckla en fristående lösning med undantag för eventuell integration med andra plattformar för t.ex. notisfunktioner eller datavisualisering.

Skalbarhet och Uppgraderingar:

- Projektet i dess nuvarande form inkluderar inte planer för framtida uppgraderingar eller skalbarhet för att utöka till fler badplatser.

Tid:

- Projektet har en fast tidsram, med specifika deadlines för varje fas. Inga aktiviteter eller utvidgningar utanför denna tidsram är planerade.

Budget:

- Projektet kommer att genomföras inom en förutbestämd budget. Utgifter som överstiger denna budget är inte planerade, och projektets omfattning kommer att anpassas för att hålla sig inom budgetgränserna.

Examensarbete

UPPFÖLJNING

Projektet förväntas att följas upp i relation till de hålltider som anges i tidsplanen. Utöver detta sker en daglig uppföljning för att löpande rapportera av projektets framsteg. Följande frågor besvaras:

Dagilga Statusrapporter, ska innehålla**:**

- Vad har jag gjort sedan igår?
- Vad ska jag göra idag?
- Är det något som hindrar mig från att göra mina uppgifter?

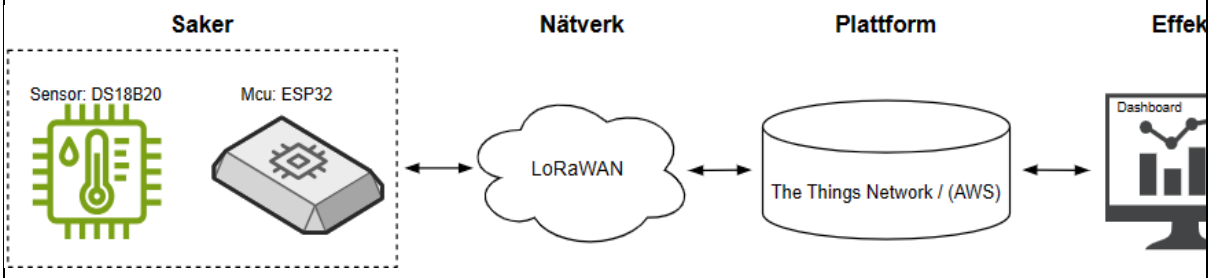
Uppföljning vid respektive hålltid, ska innehålla:

- Sammanfattning av delmomentets prestationer jämfört med planerade mål.
- Uppdateringar om eventuella avvikelser från tidsplanen och hur dessa hanteras.
- Feedback från handledare eller studentkollegor, om tillämpligt.
- Reflektioner och lärdomar från delmomentets arbete.

FÖRVÄNTAT UTFALL/MÅLUPPFYLLELSE

Lösning:

- En heltäckande IoT-lösning som omfattar en sensor, ett nätverk, en plattform och datavisualisering.



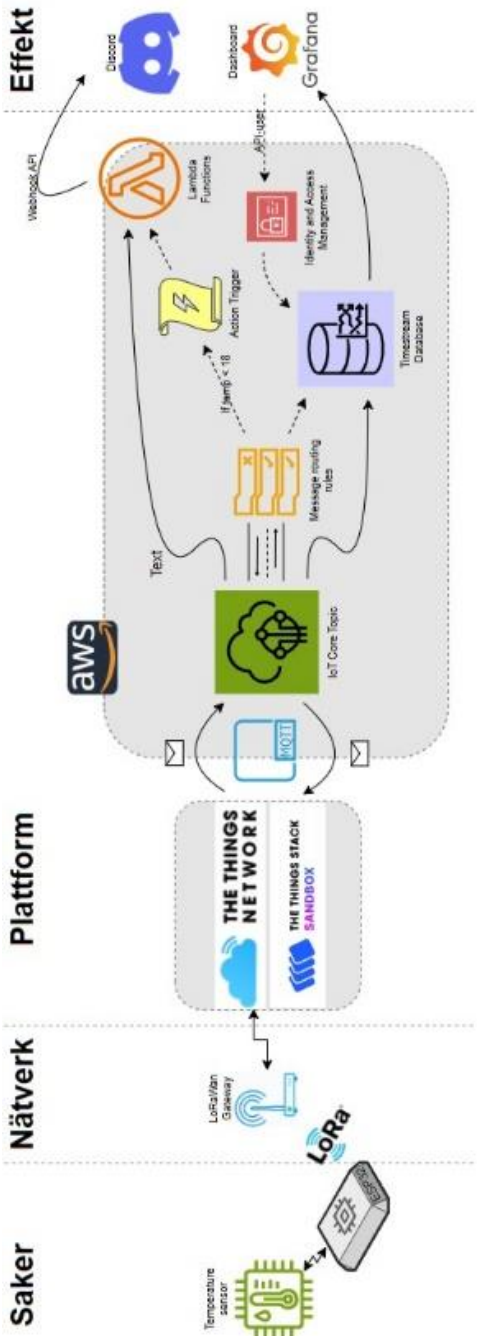
Slutprodukt:

- En digital plattform (Dashboard) för att visa vattentemperatur i realtid.

Projektrapport/Dokumentation:

- En detaljerad rapport som inkluderar projektets genomförande, tekniska specifikationer, resultat, diskussioner och slutsatser.

Bilaga 2



Figur 30 Arkitektur bilaga