# TemperaturLogger WebServer Projektbeskrivelse

## Komponenter

Projektløsningen er lavet ved brug af 2 breadboards, som er forbundet ved hjælp af et ESP32 DEVKIT V1 – DOIT board. Boardet sender strøm til breadboardet via ledningen fra 3V3 og GND portene.

En PCB-mount button bliver brugt til håndtering af reset-handlinger, og er forbundet til port 14. Knappen har en ground-forbindelse til strøm.

Der er også mountet en DS18B20 Temperature Sensor, som er den komponent som håndterer temperaturmåling. Sensoren er forbundet til VCC og til ground, og har sin Data Bus forbundet til port 4. Der er også placeret en 4.7k Ω pull-up resistor mellem VCC og data kablerne.

Et billede, der indeholder tekst, skærmbillede, elektronik, software

Indhold genereret af kunstig intelligens kan være forkert.

## Programmeringen

De nødvendige komponenters port-værdier bliver initialiseret i opstarten af TemperaturLogger-programmet. OneWire og DallasTemperature bibliotekterne er installeret for at benytte DS18B20 temperaturmåleren.

Under opstarten bliver temperaturmåleren tændt, WiFi bliver initialiseret via WiFiManager, og starter eventuelt en konfigurationsportal, hvis devicet ikke var været tilsluttet før.

Herefter bliver NTP-tid (Network Time Protocol) initialiseret, og tidszonen bliver fastsat til Københavns tidszone. NTP-tiden sikrer at vi altid får et validt tidspunkt til temperaturmålingerne.

SPIFFS bliver startet, for at opsætte filhåndtering af web-serveren.

Ved brug af while-loops sikres det at programmet venter til der er kommet gyldige data fra temperaturmåleren, og fra metoden som henter lokaltiden. Disse informationer skrives ned på en lokal csv fil på controlleren.

De forskellige http routes er opsat i programmet til håndtering af navigation på web-serveren.

Programmet sørger for at komme med en ny opdatering af temperatur, hvis man navigerer til temperature-siden.

Der bliver benyttet websocket-teknologi, som sørger for at der konstant er tovejskommunikation mellem klienten og web-serveren.

PlatformIO’s indbyggede loop funktion bliver brugt til at sørge for at reset-knappen fungerer, at WiFi-forbindelsen er OK, og at temperaturen bliver målt fast med et 5 minutters interval.

Logikken forbundet med reset-knappen er i stand til at genstarte ESP32-controlleren, web-serveren og softwaren.

Mere specifikke forklaringer på delelementer i programmet kan findes i index.html som er genereret via Doxygen.

## Opbygning af hjemmesiden

Alle handlinger på hjemmesiden styres af web-serveren fra ESP32 Devkit boardet.

For at kunne tilgå hjemmesiden, navigerer man til web-serverens IP adresse for at komme til forsiden.

De indbyggede webserver routes i TemperaturLogger programmet bliver kaldt i forbindelse med navigationer i navbaren.

Kommunikationen mellem klienten og hjemmesiden kører via WebSockets.

Hjemmesiden er opbygget med responsivitet, og har en brugervenlig visning på både håndholdte devices såvel som almindelige computerskærme.

På hjemmesiden er man i stand til at hente temperaturmåling. Når man skifter forbindelse eller mister den, vises det i konsollen. Knapperne i nav-baren giver alert-meddelelser ved handling.

I navbaren har man mulighed for at hente en csv-fil med alle nedskrevne tidsobservationer (siden filens oprettelse).

Man kan navigere til en service-side, der giver mulighed for at slette csv-filen, eller den opsatte wifi-konfiguration.

Man kan via navbaren navigere til en side, der fremviser et dynamisk view af den nuværende csv fil, hvor temperatur bliver fremvist på y-aksen, og tid på x-aksen.

Der er også mulighed for at navigere til WiFi-konfigurationssiden, der tillader klienten at ændre i det netværk, som web-serveren ligger på.

## Filoversigt

Csv\_handler.cpp

Csv\_handler.h

reset\_handler.cpp

reset\_handler.h

route\_handler.cpp

route\_handler.h

sensor\_handler.cpp

sensor\_handler.h

time\_handler.cpp

time\_handler.h

websocket\_handler.cpp

websocket\_handler.h

wifi\_handler.cpp

wifi\_handler.h

chart.js

diagram.html

index.html

output.csv

service.html

papaparse.min.js

style.css

Wificonf.html

main.cpp

config.h

Platformio.ini

.gitignore