

EH5 IoT projekt

Optimering af returtemperatur i
fjernvarmesystemer ved brug af
IoT-teknologi



Valentin-Stefan Tudora-Mayer
Studienummer: 202104813

Indhold

Introduktion	2
Projektbeskrivelse	2
Beskrivelse af systemets hovedkomponenter	3
Logikken implementeret for temperaturregulering	3
Kravs analyse.....	3
Systemdesign	5
Systemets diagram.....	5
Hardware	5
Grænseflade og kommunikation mellem komponenter	10
Implementering	10
Udviklingsprocessen for hardware og software.	10
Setup function	13
Main loop (main thread)	14
Aktuator kontrol thread	15
Weather forecast functionality.....	16
Test og Verifikation.....	18
Projekt konklusion	22
1. Projektets Resultater	22
2. Impact på Effektiviteten af Fjernvarmesystemet	22
3. Udfordringer og Lærte Lektioner	22
4. Forslag til Forbedringer og Fremtidige Udvidelser	22
5. Generelle Konklusioner	23
Referencer	24
Bilag	24
Hardware-dokumentation (skematiks, datasheets) i zip file.	24
Software kildekode.....	24

Introduktion

Dette projekt, EH5IoT-semesterprojektet, koncentrerer sig om optimering af returtemperaturen i fjernvarmesystemer for at øge energieffektiviteten og fremme bæredygtighed. Samarbejdet med Kredsløb A/S understreger vores mål om at opretholde returtemperaturer under 30°C, hvilket kvalificerer til en økonomisk bonus. Vores initiativ adresserer det voksende behov for klimavenlige løsninger i fjernvarmesektoren, et kritisk element i overgangen til mere bæredygtige energisystemer.

Målet med EH5IoT-semesterprojektet er at anvende Internet of Things (IoT)-teknologi for at præcist regulere og optimere returtemperaturen i fjernvarmesystemer. Ved at integrere avanceret sensorik og intelligent styring, sigter vi mod at forbedre systemets samlede energieffektivitet, reducere driftsomkostninger for slutbrugere og bidrage til et mere miljømæssigt ansvarligt fjernvarmesystem.

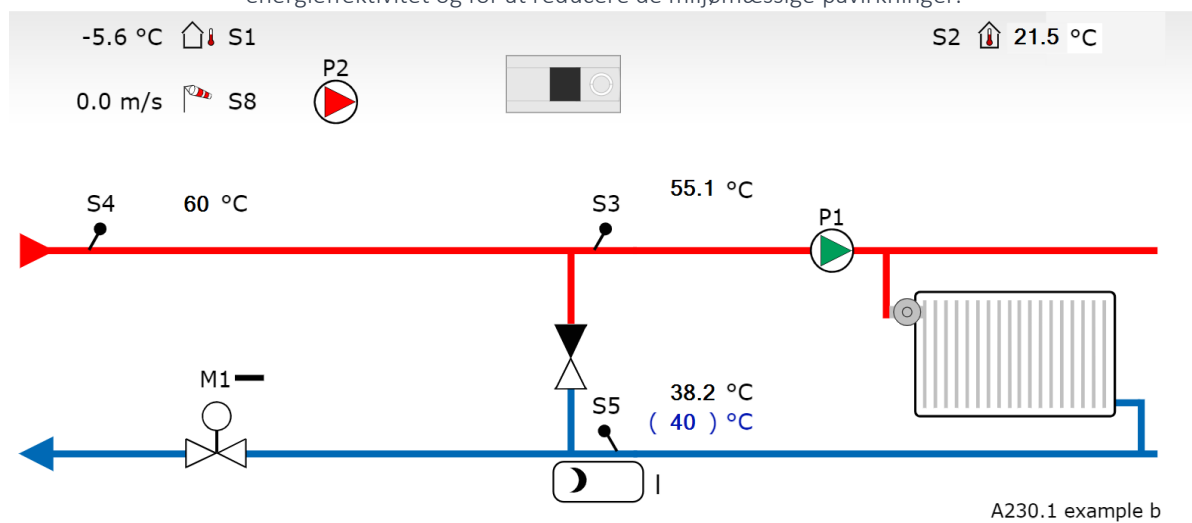
Løsningen involverer implementering af et IoT-baseret system, der bruger Particle Photon2-board til at indsamle data og styre en aktuator. Systemet vil anvende data fra vejrprognoser til dynamisk at justere setpunkterne for returtemperaturen, baseret på udendørs temperaturforhold. Ved hjælp af en RTD PT500 Kamstrup-temperatursensor monteret på returrøret, vil systemet indsamle nøjagtige data, der bruges af en PID-controller (Proportional, Integral, Derivative) til at styre en elektrisk ventil effektivt. Denne innovative tilgang tillader præcis kontrol og tilpasning af systemet i realtid, hvilket øger effektiviteten og understøtter en mere bæredygtig energianvendelse.

Projektbeskrivelse

Fjernvarme er et centraliseret varmesystem, der leverer varme til flere husstande. I dette system er returtemperaturen - temperaturen på vandet, der sendes tilbage fra boligerne til varmeværket - afgørende for systemets effektivitet. En lavere returtemperatur betyder, at mere varme er blevet udnyttet i boligerne, hvilket fører til en højere effektivitet og lavere energiforbrug.

I dette projekt, i samarbejde med Kredsløb A/S, er der et specifikt fokus på at opretholde returtemperaturen på 30°C, mens indgangstemperaturen, der garanteres af Kredsløb A/S, er på 60°C (Figur 1). Dette temperaturinterval er afgørende for at sikre maksimal udnyttelse af den leverede varmeenergi. For eksempel, en reduktion i returtemperaturen fra 35°C til 30°C, mens indgangstemperaturen forbliver på 60°C, betyder en forbedring i energieffektiviteten.

Optimeringen af returtemperaturen er derfor afgørende for at forbedre fjernvarmesystemets overordnede energieffektivitet og for at reducere de miljømæssige påvirkninger.



Figur 1 [1]Fjernvarme flowdiagram (Kilder - Kredsløb A/S - ECL portal)

Beskrivelse af systemets hovedkomponenter

Systemet blev udviklet til dette formål og omfatter flere nøglekomponenter:

1. **Particle Photon2:** Dette er en mikrocontroller, der fungerer som kernen i IoT-løsningen. Den indsamler data fra forskellige sensorer og styrer aktuatorer baseret på foruddefinerede logikker og eksterne datainput. En afgørende funktion ved Particle Photon2 er dens evne til at forbinde til internettet, hvilket muliggør fjernovervågning og -styring af systemet. Takket være Particles integrerede cloud-tjenester, kan systemet nemt tilgå og anvende data fra eksterne kilder, såsom vejrprognoser, hvilket spiller en nøglerolle i projektet. Denne forbindelse til internettet udvider systemets funktionalitet betydeligt og tillader en mere fleksibel og effektiv styring af fjernvarmesystemet.
2. **Temperatursensor PT500 Kamstrup:** Denne sensor er monteret på returrøret og måler returtemperaturen med præcision. Disse data er afgørende for at regulere og vedligeholde den optimale returtemperatur.
3. **Aktuator (24V Elektrisk Ventil M3P 46825-30):** Denne bruges til at justere vandstrømmen og dermed returtemperaturen baseret på data og kommandoer fra Particle Photon2 og PID-controlleren.

Logikken implementeret for temperaturregulering.

Systemet anvender en dynamisk logik til temperaturregulering, som er afhængig af eksterne temperaturforhold. En PID-controller (Proportional, Integral, Derivative) anvendes, som justerer ventilens position for at opnå den ønskede returtemperatur. Controlleren modtager input fra temperatursensoren og justerer aktivt ventilen baseret på tre parametre:

- **Proportional (P):** Justerer den aktuelle fejl.
- **Integral (I):** Akkumulerer tidligere fejl for at forbedre stabilitet over tid.
- **Derivative (D):** Forudsiger fremtidige fejl og reagerer på ændringer.

Ved at integrere realtidsdata fra sensorer og vejrprognoser, kan systemet proaktivt justere set punkterne for returtemperaturen, hvilket sikrer maksimal effektivitet under alle driftsforhold.

Kravs analyse

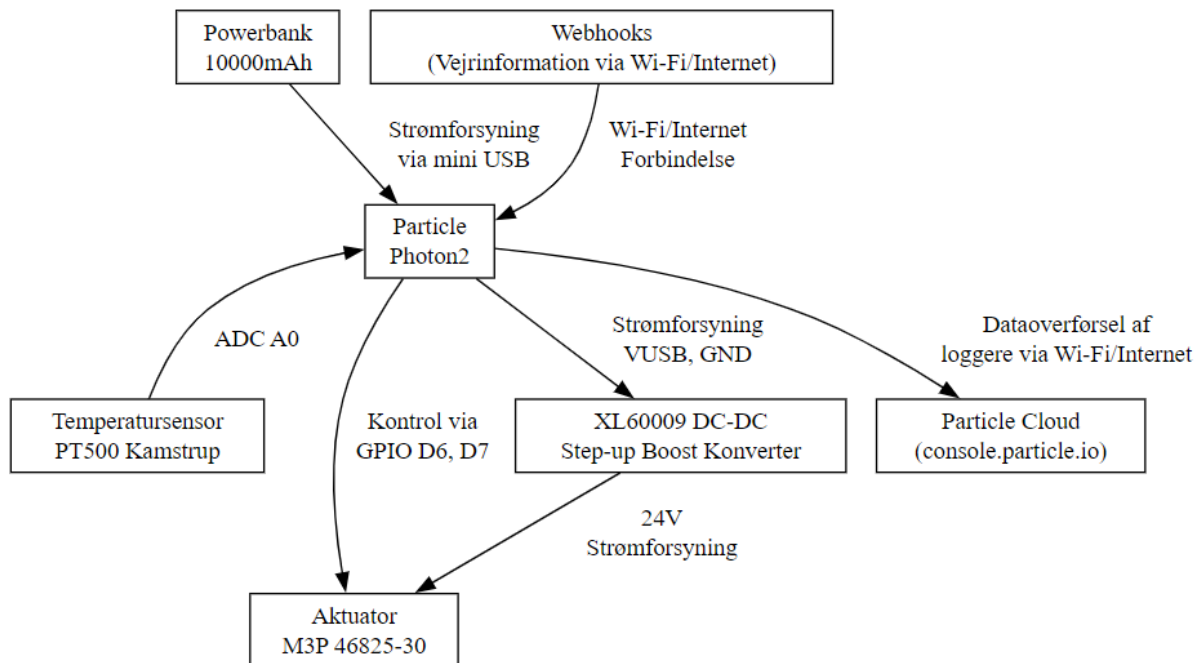
Krav ID	Kategori	Beskrivelse	Godkendt/Ikke godkendt	Kommentarer
F.1	Funktionelt	Enheden skal forbinde til internettet via WiFi.	Godkendt	AU's "AU Gadget netværk" er testet og fungerer.
F.2	Funktionelt	Enheden skal aflæse data fra PT500-sensoren, som måler temperatur og implementerer datatiltrering.	Godkendt	PT500 sensoren er implementeret og funktionalitet bekræftet via tests.
F.3	Funktionelt	Enheden skal kunne styre en aktuator, som er reguleret af en PID-controller.	Godkendt	PID-reguleringen er implementeret og justeret baseret på tests.

Krav ID	Kategori	Beskrivelse	Godkendt/Ikke godkendt	Kommentarer
F.4	Funktionelt	Enheden skal integrere og anvende data fra eksterne webtjenester til at forbedre dens funktionalitet.	Godkendt	Integration af webhooks og API'er for vejrdato er testet og implementeret.
F.5	Funktionelt	Data skal sendes til clouden i en log med defineret interval og størrelse for at monitorere systemets ydeevne.	Godkendt	Data logging og transmission er implementeret og fungerer som forventet.
F.6	Funktionelt	Aktuatoren skal kunne modtage strøm fra en batterikilde ved hjælp af en DC-DC boost step-up konverter.	Godkendt	Implementeringen og tests af DC-DC konverteren har bekræftet korrekt funktion.
IF.1	Ikke-funktionelt	Offentlig GitHub-repository med alle relevante filer.	Godkendt	Repository er aktivt og offentligt tilgængeligt.
IF.2	Ikke-funktionelt	Hardware dokumentation inklusiv skematikker og datablade skal uploades i PDF-format.	Godkendt	Komplet dokumentation er tilgængelig og godkendt.
IF.3	Ikke-funktionelt	Software kildekoden skal uploades i rå format.	Godkendt	Kildekoden er tilgængelig i repository og er blevet gennemgået.
T.1	Teknisk	Den indlejrede platform skal være en Particle-enhed med WiFi-forbindelse.	Godkendt	Particle-enhed valgt og testet.
T.2	Teknisk	Platformen skal have digitale/analoge I/O'er for sensor- og aktuatorforbindelser.	Godkendt	I/O-specifikationer er opfyldt og fungerer som forventet.
T.3	Teknisk	Systemet skal kunne drives af batteri, ud over USB 5V strømforsyning.	Godkendt	Batteridrift er testet og klar til yderligere overvågning for at optimere levetiden.
T.4	Teknisk	Batteriets levetid og systemets samlede energiforbrug skal evalueres for at estimere driftstiden ved fuld batteriopladning.	Godkendt	Yderligere langvarige tests og overvågning nødvendige for at fastlægge nøjagtig batterilevetid.

Bemærk, at selvom alle kravene er markeret som "Godkendt", indikerer kommentaren for T.4, at der stadig er behov for yderligere langvarige tests og overvågning for præcist at bestemme batteriets levetid, især fordi aktuatoren har et højere strømforbrug under drift, hvilket er kritisk for den overordnede energieffektivitet og anvendelighed af systemet i områder uden elektrisk strømforsyning.

Systemdesign

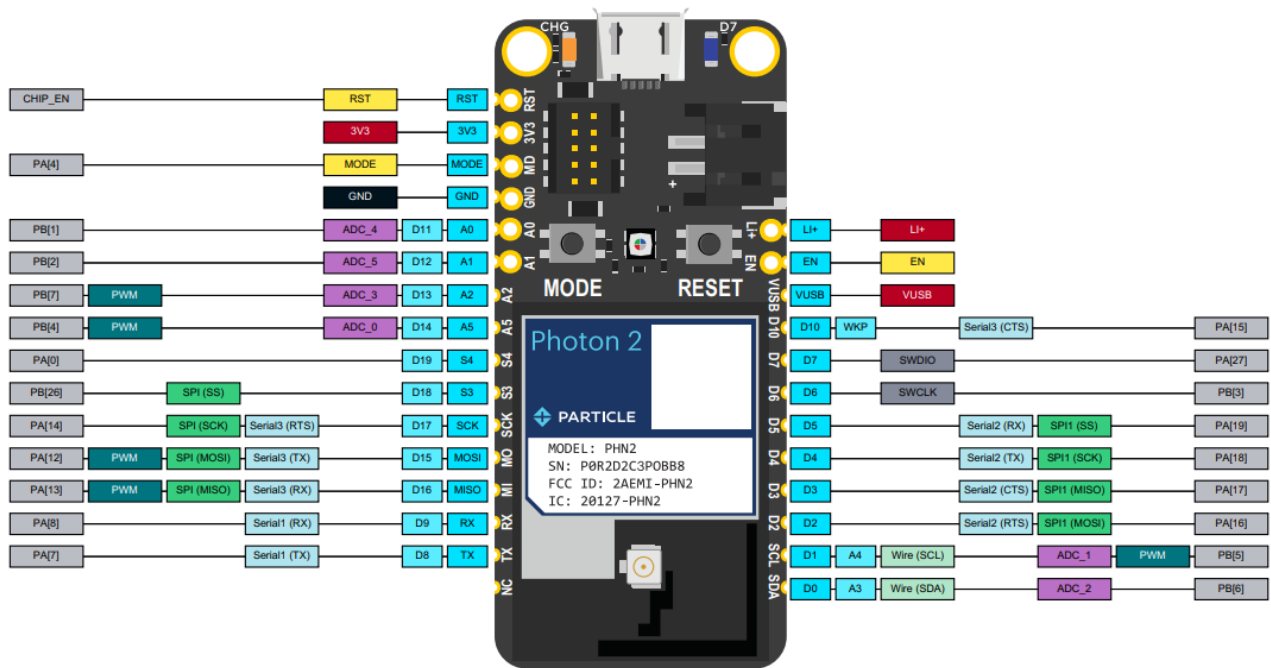
Systemets diagram.



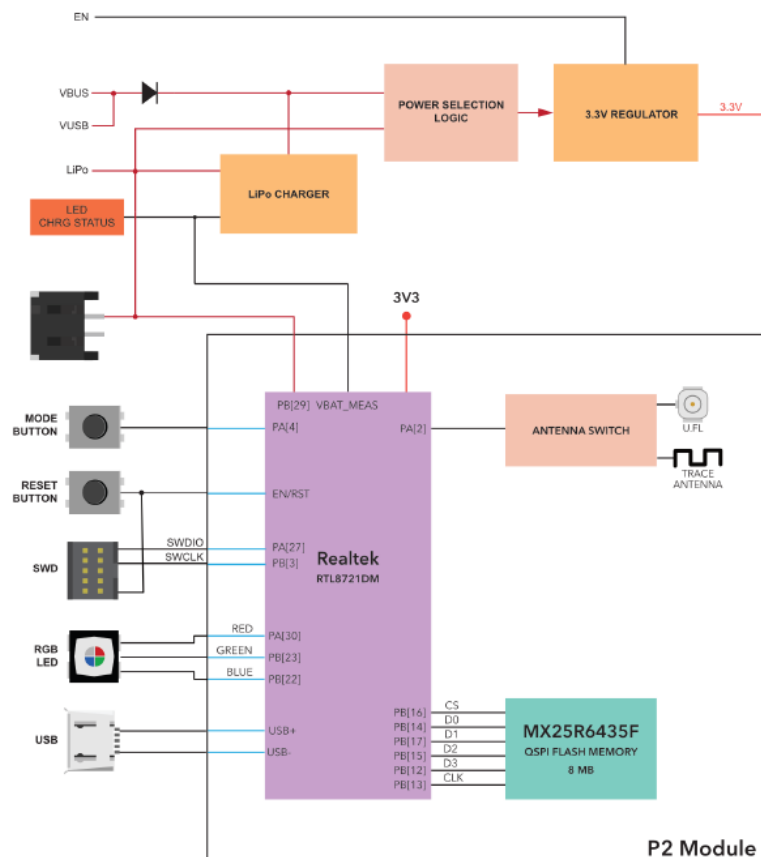
Figur 2 System Block Diagram

Hardware

- **Particle Photon2** (Figur 3 og Figur 4 [2]) er en avanceret mikrocontroller modul designet til Internet of Things (IoT) applikationer som tilbyder :
 - **MCU & Processor:** Realtek RTL8721DM, ARM Cortex M33, 200 MHz.
 - **Hukommelse:** 2 MB Flash, 3 MB RAM.
 - **Wi-Fi:** Dual-band 2.4 GHz & 5 GHz, BLE 5 support.
 - **Antenne:** Integreret PCB, U.FL for ekstern antenne.
 - **I/O Porte:** 6x ADC, 3x UART, 2x SPI, I2C-kommunikation.
 - **Strømstyring:** Understøtter LiPo-batteri, USB-strøm.
 - **Lavenergimodes:** Effektive sleep modes for strømbesparelse.
 - **Firmware:** Over-the-air opdateringer, Device OS 5.0.0+.



Figur 3 [3](001)Photon2 pinouts

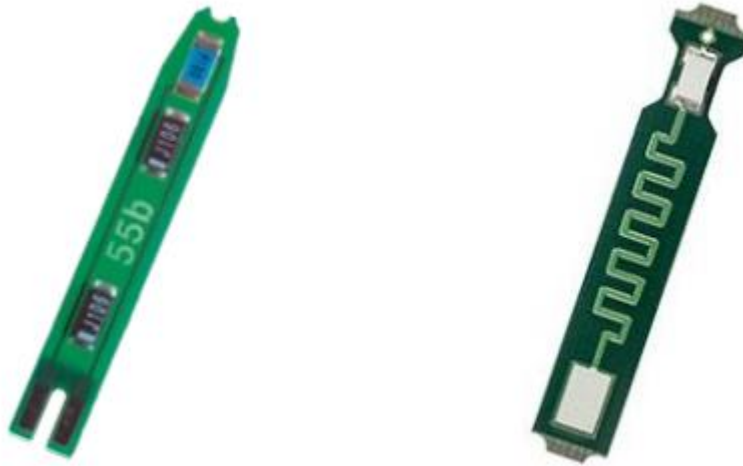


Figur 4 [3] (001)Photon2 - Block Diagram

- **PT500** er en 2-leder resistiv temperaturføler (Figur 6), anvendt til nøjagtig måling af temperaturen i fjernvarmesystemer. PT500 temperaturføleren, ofte anvendt i varme- og kølesystemer, bruger tyndfilmsmodstande monteret på et trykt kredsløbskort (PCB). Tyndfilmsmodstande er valgt for deres omkostningseffektivitet, kompakthed og hurtige respons på temperaturændringer. Denne konstruktion tillader masseproduktion og reducerer omkostninger. Med loddeøer på PCB, er det let at forbinde to ledninger til en 2-leder temperaturføler, hvilket optimerer fremstillingsprocessen. PT500 sensoren kan typisk bruges op til 150 °C, hvilket er tilstrækkeligt for de fleste installationer.

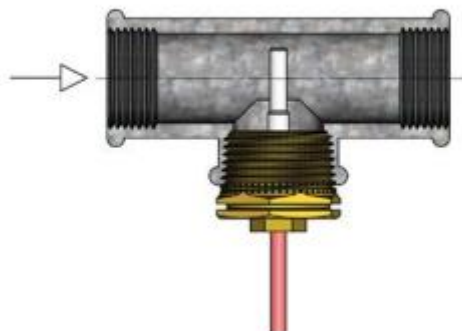
Sensoren har følgende karakteristika:

- Modstandsændring: Sensoren benytter platin med en temperaturkoefficient på 0,00385 ohm/ohm/°C, hvilket sikrer en præcis og næsten lineær respons til temperaturændringer.
- Tabelbaseret beregning: Til konvertering af modstandsværdier til temperaturer anvendes en standardtabel, som er integreret i systemets software. Dette sikrer en høj grad af nøjagtighed i temperaturbestemmelsen.
- 2-leder konfiguration: Selvom PT500 også er tilgængelig i 4-leder konfigurationer for endnu større nøjagtighed, anvendes en 2-leder konfiguration i dette projekt



Figur 5 [3](002)Tyndfilmsmodstande, som er monteret på et PCB (printed circuit board)

Direkte temperaturføler

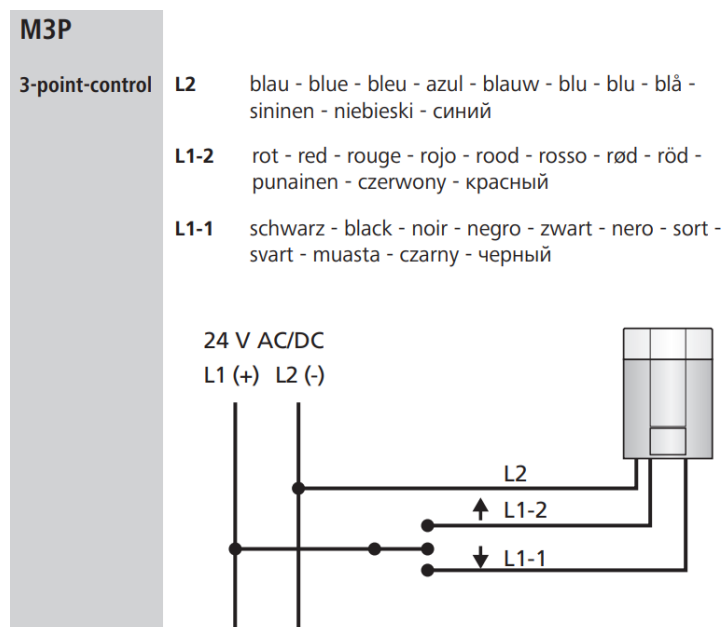


Figur 6 [3](002)Illustration af en direkte temperaturføler

- **M3P 46825-30 aktuatoren** (Figur 7) er en motoriseret aktuator til regulering af flow i varme- og kølesystemer.

Tekniske specifikationer inkluderer:

- Kontroltype: 3-punkts kontrol uden feedback.
- Driftsspænding: 24 V AC, -10 % ... +20 %, 50-60 Hz 24 V DC, -20 % ... +20 %
- Driftseffekt: 0,2 W...1,4 W (DC), 0,56 VA...2,6 VA (AC)
- Strømforbrug (M3P): < 110 mA
- Indstillingsvej: 8,5 mm
- Indstillingskraft: 100/125/150/200 N -20 % ... +40 %
- Indstillingstid: 15s, 30s s/mm
- Omgivelsestemperatur: 0 °C ... +50 °C
- Beskyttelsesklasse: IP 54 i henhold til EN 60529.
- Kontrolsignal: 3-punkts signal.
- Denne aktuator er central i projektet for at præcist justere ventilpositionen, hvilket direkte påvirker returtemperaturen i opvarmningssystemet, og bidrager til at forbedre systemets energieffektivitet.



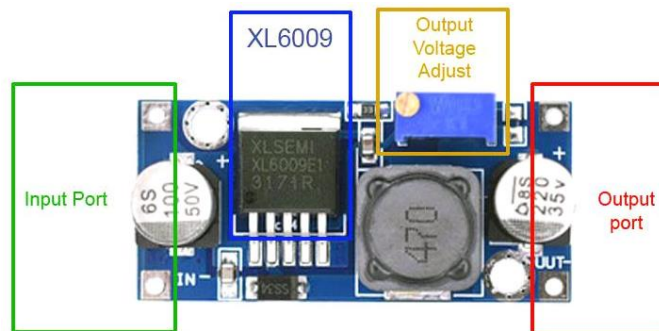
Figur 7 [3](003) Elektrisk diagram

- **Powerbanken PB20A** (Figur 8), anvendt i projektet, til at forsyne med strøm, har følgende nøgleegenskaber:
 - Kapacitet: 10.000 mAh, 3,7 V, 37 Wh, hvilket sikrer tilstrækkelig strøm til langvarig brug af systemet.
 - Indgang og Udgang: Udstyret med USB-C-indgang (5 V/2 A) og USB-A-udgang (5 V/2 A), der tilbyder alsidige opladningsmuligheder og høj kompatibilitet med forskellige enheder.
 - Sikkerhedsfunktioner: Inkluderer overladningsbeskyttelse og automatisk slukning efter fuld opladning, hvilket sikrer både powerbankens og tilkoblede enheders sikkerhed.



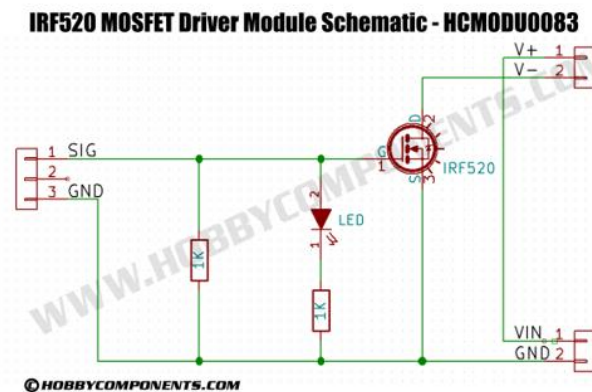
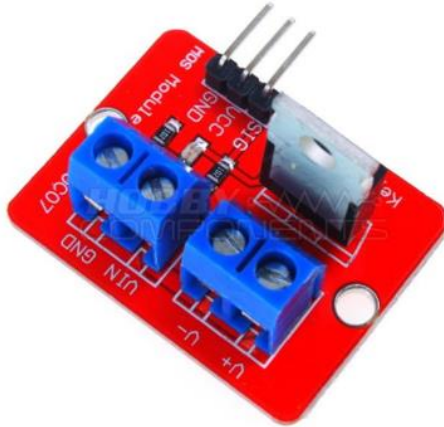
Figur 8 [3](004) Powerbank

- **XL6009 DC-DC** (Figur 9): Specifikationer og egenskaber for strømmodul til at levere strøm til en 24V aktuator:
 - **Indgangsspænding:** 3 - 32V.
 - **Udgangsspænding:** Justerbar fra 5 - 35V.
 - **Udgangsstrøm:** Maksimal udgangsstrøm 4A. Bemærk: Jo højere spændingen er, desto større bliver belastningsstrømmen.
 - **Effektivitet:** Op til <94% effektivitet.
 - **Belastningsregulering:** 0,5%.
 - **Spændingsregulering:** 0,5%.
 - **Justerbar potentiometer:** Ombord for justering af udgangsspændingen.



Figur 9 [3] (005) XL6009 DC-DC Convertor module

- **IRF520** (Figur 10) Power MOSFET N-Channel modulen benytter en IRF520 Power MOSFET for at styre og regulere PWM-signaler til belastninger som DC-motorer og magnetventiler.
 - Modulen kan håndtere en udgangsspænding på 0-24V og en udgangsstrøm på op til 9,5A. For strømme over 1A kræves en ekstra køleplade. Den understøtter gate-spændingssignaler på både 3,3V og 5V og er forsynet med digitale porte til niveau-signaler.



Figur 10 [4](006) IRF520 Modulen og Skematik

Grænseflade og kommunikation mellem komponenter.

Implementering

Udviklingsprocessen for hardware og software.

Implementeringsfasen i EH5IoT-projektet omhandler integration og konfiguration af hardware og softwarekomponenter for at skabe et fungerende IoT-system som skal afleveres en retur temperatur som er indstillet på setpoint.

Det inkluderer etablering af forbindelser mellem komponenter som Particle Photon2, PT500 temperatursensoren, M3P 46825-30 aktuatoren, powerbanken, XL6009 DC-DC konverteren og IRF520 Power MOSFET. Under denne fase er der lagt vægt på at sikre korrekt energioverførsel, dataindsamling og -behandling samt effektiv kommunikation mellem systemets dele. Desuden er der udviklet software til at styre og regulere systemet baseret på realtidsdata og prædefinerede algoritmer.

Prototypen til projektet blev udviklet på et eksperimentbræt (breadboard), hvor en PCB ikke var nødvendig. Systemets strømforsyning kommer fra en powerbank, der leverer 5V via USB til Particle Photon2-plattformen. Fra Particle Photon2's Vusb-pin leveres strøm til XL6009 DC-DC-modulet, som konverterer spændingen op til de nødvendige 24V for aktuatoren M3P 46825-30. Alternativt kunne en SAFT longlife-batteri bruges, der ofte anvendes i fjernvarmeapplikationer og industrielle miljøer.

Aktuatoren styres digitalt fra Photon2's pins D6 og D7, via to IRF520 Power MOSFET-moduler, som understøtter gate-spændingssignaler på 3,3V og 5V og kan håndtere en udgangsspænding på 0-24V og en udgangsstrøm på op til 9,5A. Temperaturen setpoint styres dynamisk baseret på vejrprognoser hentet via en webhook, med sætpunkter på 30, 35 eller 40 °C.

Temperature Condition	Set Point
Extreme low temperature (< -3°C)	40°C
Low temperature (-2°C to 2°C)	35°C
Normal temperature (> 3°C)	30°C

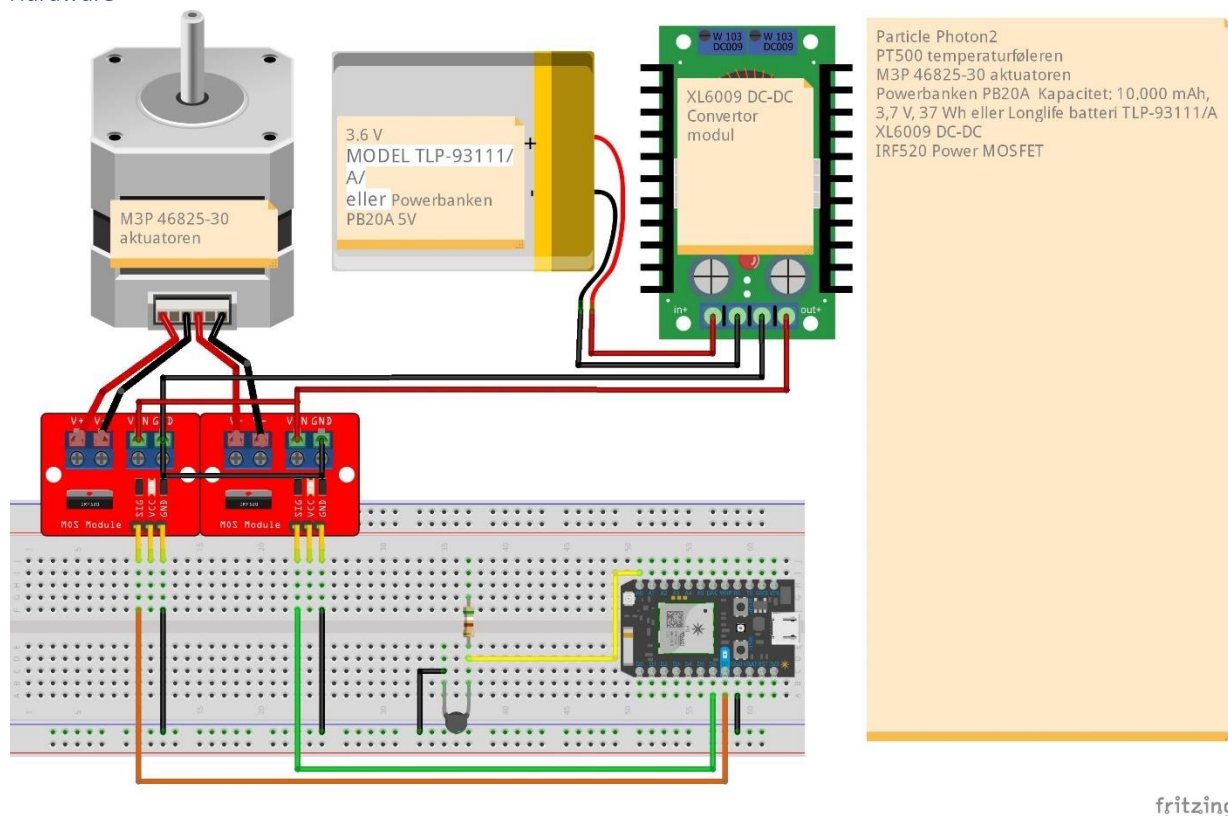
Tabel 1 Setpoint temperature

Returtemperaturen registreres ved hjælp af en PT500 temperatursensor, som er tilsluttet mellem A0-pinden på en Photon2, GND, og en 1k Ohm variabel modstand. Dette setup gør det muligt at beregne resistansen af PT500-sensoren. PID-regulatoren tilpasser herefter aktuatoren bevægelserne for at holde den ønskede temperatur, baseret på feedback fra PT500-sensoren. Værdien af Pt500 beregnes ved brug af en tabel (Tabel 2) over resistansværdier, som svarer til forskellige temperaturer, leveret af sensorens producent, Kamstrup A/S.

Pt500										
°C	0	1	2	3	4	5	6	7	8	9
0	500,000	501,954	503,907	505,860	507,812	509,764	511,715	513,665	515,615	517,564
10	519,513	521,461	523,408	525,355	527,302	529,247	531,192	533,137	535,081	537,025
20	538,968	540,910	542,852	544,793	546,733	548,673	550,613	552,552	554,490	556,428
30	558,365	560,301	562,237	564,173	566,107	568,042	569,975	571,908	573,841	575,773
40	577,704	579,635	581,565	583,495	585,424	587,352	589,280	591,207	593,134	595,060
50	596,986	598,911	600,835	602,759	604,682	606,605	608,527	610,448	612,369	614,290
60	616,210	618,129	620,047	621,965	623,883	625,800	627,716	629,632	631,547	633,462
70	635,376	637,289	639,202	641,114	643,026	644,937	646,848	648,758	650,667	652,576
80	654,484	656,392	658,299	660,205	662,111	664,017	665,921	667,826	669,729	671,632
90	673,535	675,437	677,338	679,239	681,139	683,038	684,937	686,836	688,734	690,631
100	692,528	694,424	696,319	698,214	700,108	702,002	703,896	705,788	707,680	709,572
110	711,463	713,353	715,243	717,132	719,021	720,909	722,796	724,683	726,569	728,455
120	730,340	732,225	734,109	735,992	737,875	739,757	741,639	743,520	745,400	747,280
130	749,160	751,038	752,917	754,794	756,671	758,548	760,424	762,299	764,174	766,048
140	767,922	769,795	771,667	773,539	775,410	777,281	779,151	781,020	782,889	784,758
150	786,626	788,493	790,360	792,226	794,091	795,956	797,820	799,684	801,547	803,410
160	805,272	807,133	808,994	810,855	812,714	814,574	816,432	818,290	820,148	822,004
170	823,861	825,716	827,571	829,426	831,280	833,133	834,986	836,838	838,690	840,541
180	842,392	844,241	846,091	847,940	849,788	851,635	-	-	-	-

Tabel 2 [3] - Pt500, IEC 751 Amendment 2-1995-07

Hardware



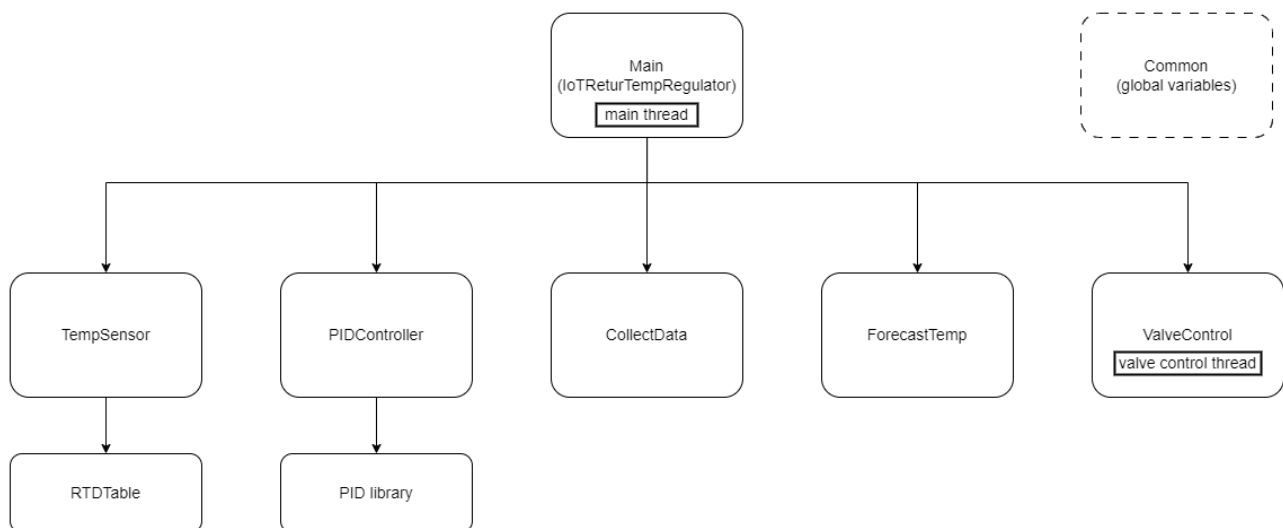
Figur 11 Implementering skematik

Software

Softwarekomponenterne i projektet arbejder sammen for at regulere og overvåge fjernvarmesystemets returtemperatur. Kernen i softwarelogikken er en PID-controller, som dynamisk justerer en aktuator position baseret på feedback fra en PT500 temperatursensor. Temperaturen læses og jævnes ud gennem en glidende gennemsnitsfunktion for at sikre stabil drift.

Vigtige funktioner i softwaren omfatter indsamling af temperaturdata, kommunikation med vejrprognoser og systemets interaktion med Particle Cloud for at logge og overvåge ydeevnen. PID-controllerens konstanter (K_p , K_i , K_d) er defineret for at balancere systemets respons mellem nuværende, akkumuleret og forventet temperaturafvigelse. Systemet håndterer også intervalbaserede opgaver som indsamling og offentliggørelse af data samt opdatering af setpoint temperature baseret på vejrforudsigelser som er beskrevet i *Tabel 1*.

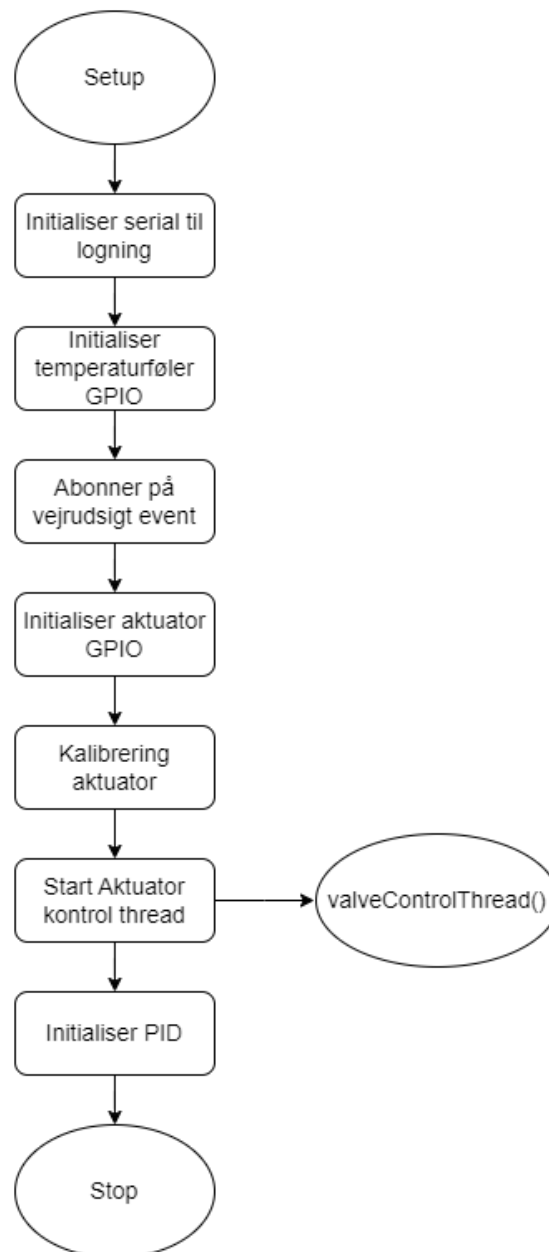
Hele implementeringen er trådbaseret for at sikre kontinuerlig og uafhængig håndtering af de forskellige systemopgaver. Koden er skrevet i C++ og gør brug af Particle's udviklingsmiljø og biblioteker. For detaljeret gennemgang og yderligere teknisk indsigt, kan koden tilgås på GitHub [5].



Figur 12 Forsimplet Software Diagram

Setup function

På diagrammet kan man se opsætningsproceduren, som skal køres før man går ind i hovedløkken. Vi initialiserer serial kommunikation til brug for logning. Logning er blevet brugt omfattende gennem udviklingsprocessen til fejlfinding. Particle understøtter flere logniveauer, og programmet er i øjeblikket indstillet til at logge alle INFO beskeder. Dernæst initialiseres GPIO'erne for både temperatursensoren (som analog indgang) og aktuatoren (som digital udgang). Vi abonnerer på vejr-begivenheden, og navnet på begivenheden vil senere blive brugt af Particle Cloud til at sende data tilbage. Her specificerer vi også callback-funktionen til at fortolke de modtagne data. Aktuatoren kalibreres derefter, dette er for at sikre, at vi starter programmet fra en kendt position af aktuatoren. Aktuatoren vil bevæge sig helt op (100% åben) og lukke ned til frostbeskyttelsespositionen (5%). Denne proces er tidskrævende, kan tage op til 4 minutter. Af denne grund tilføjede jeg et debug-flag for at deaktivere kalibrering under udviklingen. Til sidst starter vi valveControlThread, så den vil være klar til at bruge outputtet fra PID-controlleren. PID-controlleren initialiseres også her: outputområdet (0-100), det indledende setpunkt (30) og funktionsmåden (Automatisk).

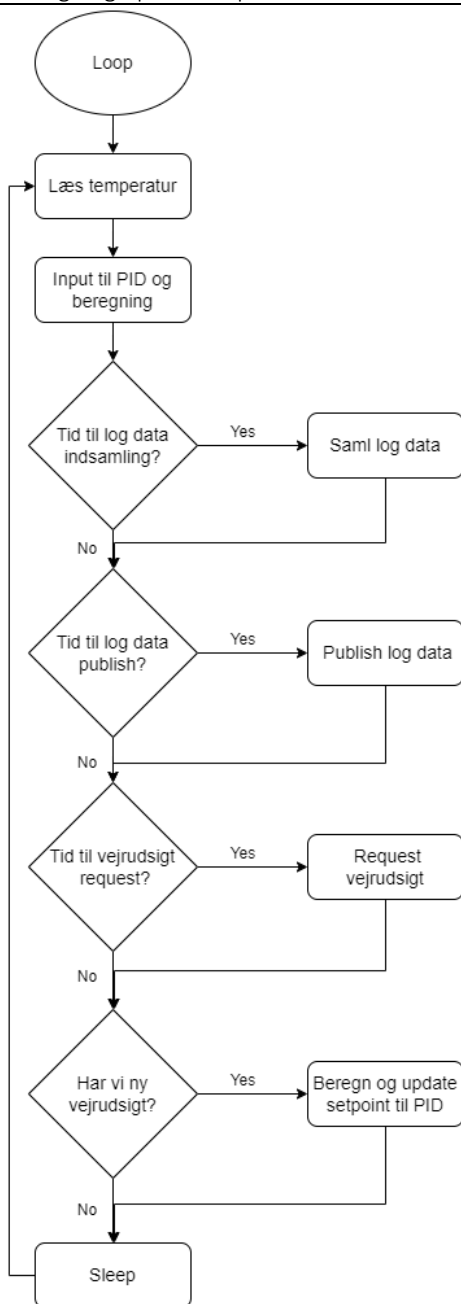


Figur 13 Setup Flow Chart

Main loop (main thread)

Funktionaliteten af hovedtråden er ret simpel, da den kun læser temperatursensoren og giver dette som input til PID-controlleren. Vi gemmer resultatet af PID-controllerens beregning i en global variabel, som skal bruges i den anden tråd. Af denne grund er vi nødt til at bruge en mutex og låse den, før vi manipulerer med variabelen. Resten af funktionaliteten i hovedløkken består af logning af data, at sende disse data til skyen, at anmode om vejrudsigt og bruge udsigten til at konfigurere setpunktet. Alt dette sker ved foruddefinerede tidsintervaller, som ses i følgende tabel. Til sidst har vi en forsinkelsesfunktion på 1 sekund.

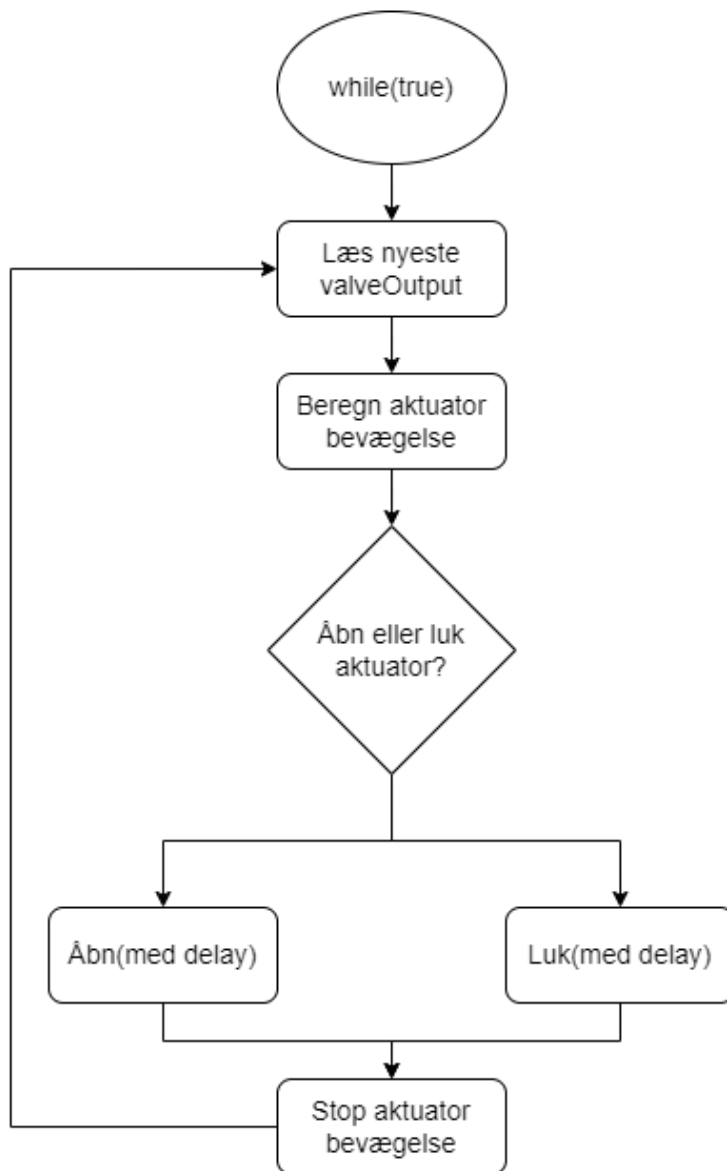
Aktion	Tids interval
Saml log data	1 time
Publish log data	3 timer
Request vejrudsigt	3 timer
Beregn og update setpoint til PID	Dataen skal være nyere end 3 timer



Figur 14 Main Loop Flow Chart

Aktuator kontrol thread

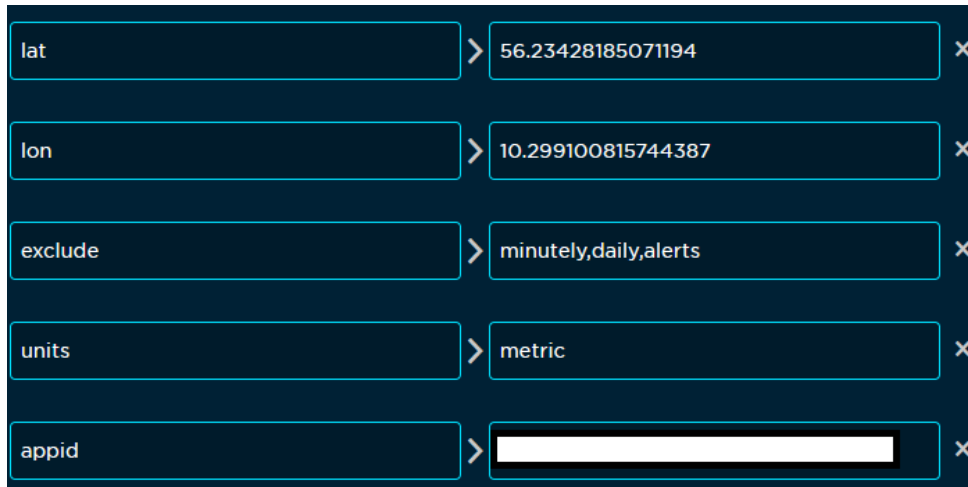
Den anden tråd i applikationen håndterer styringen af aktuatoren. Aktuatorernes nuværende position er gemt i en variabel. Vi læser den globale variabel for den nye position beregnet af PID-controlleren. Og vi beregner forskellen mellem de to for at finde ud af, hvilken retning aktuatoren skal bevæge sig. Baseret på dette vil vi enten åbne eller lukke aktuatoren ved hjælp af de to digitale GPIO'er. Problemet er dog, at denne proces er ret langsom. Dette var en af de væsentligste grunde til at flytte aktuatorfunktionaliteten til en separat tråd – for at undgå at have en stor mængde forsinkelser i hovedtråden. Hvis PID-controlleren siger, at aktuatoren ikke behøver at bevæge sig, vil tråden sove i et sekund.



Figur 15 Valve_control_thread_flow_chart

Weather forecast functionality

Kommunikationen med openweathermap API foretages af Particle Cloud, ved hjælp af deres Integrations-funktionalitet. Vi har brug for timebaserede data, så dette skulle specificeres i forespørgslen til openweathermap. Dette kan gøres i konfigurationen af Integrationen, hvor du også skal specificere placeringen, måleenhederne og API-nøglen.

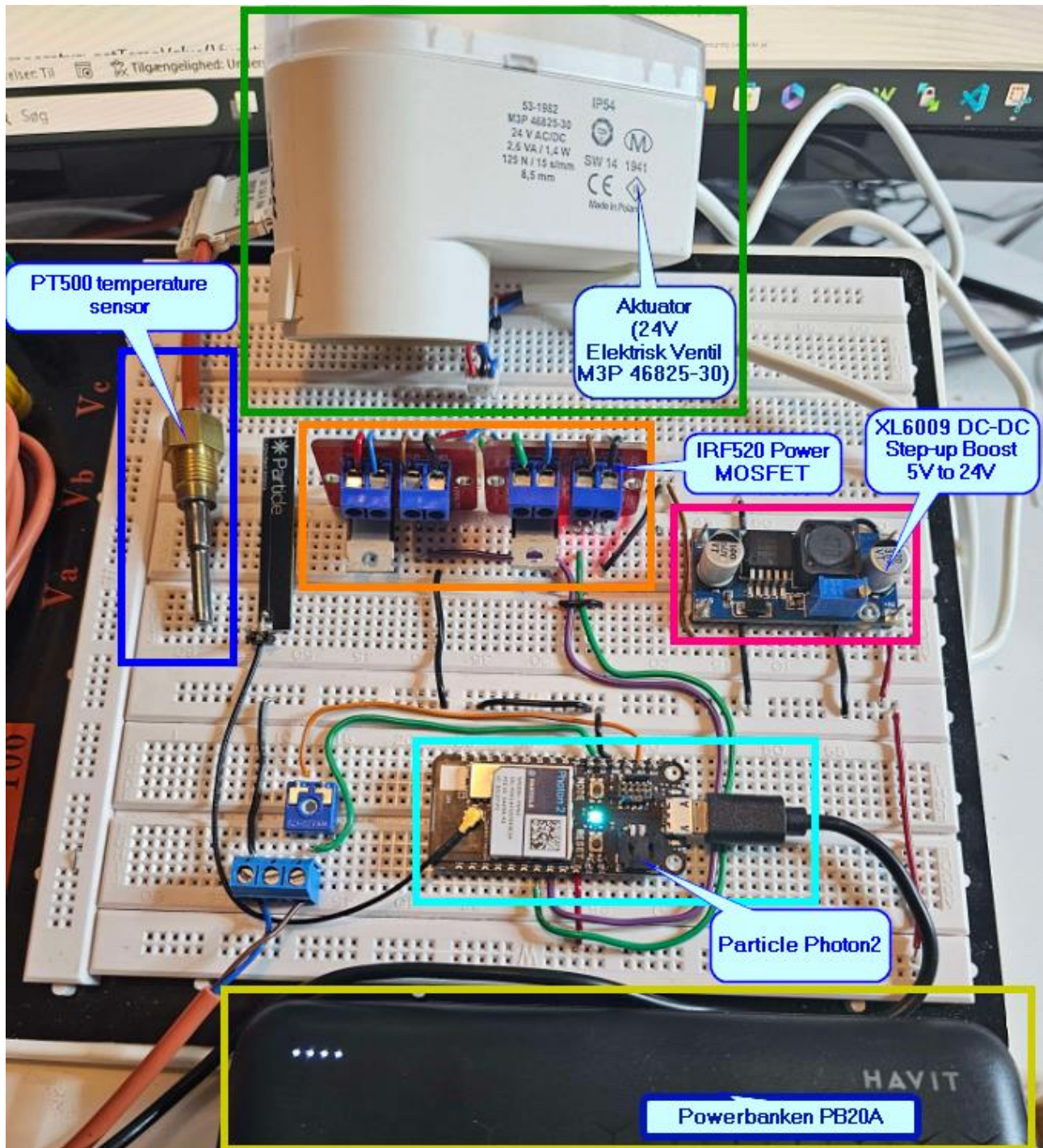


En meget fin funktion ved Particle Integration er, at den kan filtrere dataene for os. Ved det første forsøg på at håndtere data direkte fra openweathermap indse jeg, at jeg ikke kunne håndtere mængden af data, der blev sendt (48 datasæt for de følgende 48 timer). Følgende "Response Template" bruges til kun at filtrere de data, vi er interesserede i (de næste 3 timer).

Response Template ⓘ

```
{"current": {"dt": "{{current.dt}}", "temp": "{{current.temp}}"}, "hourly": [{"dt": "{{hourly.0.dt}}", "temp": "{{hourly.0.temp}}"}, {"dt": "{{hourly.1.dt}}", "temp": "{{hourly.1.temp}}"}, {"dt": "{{hourly.2.dt}}", "temp": "{{hourly.2.temp}}"}]}
```

Når jeg modtager dataene på Particle Photon, har jeg brugt en funktion genereret på <https://docs.particle.io/integrations/webhook-demo/> til at dekode JSON-objekterne.



Figur 16 Realisering af IoT prototype på breadboard

Test og Verifikation

```
0000123249 [app] INFO: Current step: 95
0000124527 [app] INFO: Current step: 96
0000125806 [app] INFO: Current step: 97
0000127084 [app] INFO: Current step: 98
0000128363 [app] INFO: Current step: 99
0000129641 [app] INFO: Current step: 100
0000129645 [app] INFO: Stop valve...
0000129648 [app] INFO: Step difference to move valve: -95
0000129652 [app] INFO: Closing valve...
0000130931 [app] INFO: Current step: 99
0000132209 [app] INFO: Current step: 98
0000133488 [app] INFO: Current step: 97
0000134766 [app] INFO: Current step: 96
0000136045 [app] INFO: Current step: 95
```

Figur 17 Test calibration - valve change direction

```
0000244725 [app] INFO: Current step: 10
0000246005 [app] INFO: Current step: 9
0000247283 [app] INFO: Current step: 8
0000248562 [app] INFO: Current step: 7
0000249841 [app] INFO: Current step: 6
0000251119 [app] INFO: Current step: 5
0000251122 [app] INFO: Stop valve...
0000251126 [app] INFO: Step difference to move valve: 0
```

Figur 18 Test calibration finish antifreeze position

```
0000252303 [app] INFO: Event (GetWeatherForecast) data received
0000252310 [app] INFO: Weather forecast, currentTime: 1703191022, currentTemp 4.610000 C
0000252318 [app] INFO: Weather forecast, hour 1 Time: 1703188800, hour 1 Temp 4.480000 C
0000252326 [app] INFO: Weather forecast, hour 2 Time: 1703192400, hour 2 Temp 4.610000 C
0000252334 [app] INFO: Weather forecast, hour 3 Time: 1703196000, hour 3 Temp 4.380000 C
```

Figur 19 Test calibration forecast Particle



Figur 20 Test calibration forecast Cloud

```
0000531645 [app] INFO: Current Resistance: 550.91, temperature 26.15, averaged temperature: 26.10
0000531654 [app] INFO: Input: 26.10, Setpoint: 30.00, Temp Diff: 3.90, Valve Output: 65.13
0000531662 [app] INFO: Added data to log: timestamp: 1703191301, temperature: 26.10, valveOutput: 65.13
```

Figur 21 Test individual data log save

```
0000136592 [app] INFO: Event (SendCollectedData) data sent: {"node":"iot10","deviceId":"0a10aced202194944a049c24","valveOutput":~
0000136612 [app] INFO: Collected data, timestamp 1703192462, temperature 25.264425 C valveOutput34.184996
0000136621 [app] INFO: Collected data, timestamp 1703192502, temperature 25.153606 C valveOutput43.426546
0000136630 [app] INFO: Collected data, timestamp 1703192542, temperature 25.375246 C valveOutput50.745482
```

Figur 22 Test data log Particle

SendCollectedData

Published by **Oa10aced202194944a049c24** on **12/21/23** at **10:02:23 pm**

PRETTY

RAW

```
{
  "node": "iot10"
  "deviceId": "Oa10aced202194944a049c24"
  "data": [
    0: {
      "timestamp": 1703192462
      "temperature": "25.26"
      "valveOutput": "34.18"
    }
    1: {
      "timestamp": 1703192502
      "temperature": "25.15"
      "valveOutput": "43.43"
    }
    2: {
      "timestamp": 1703192542
      "temperature": "25.38"
      "valveOutput": "50.75"
    }
  ]
}
```

Figur 23 Test data log Cloud

```

0000170605 [app] INFO: Current Resistance: 564.54, temperature 33.19, averaged temperature: 27.79
0000170614 [app] INFO: Input: 27.79, Setpoint: 30.00, Temp Diff: 2.21, Valve Output: 43.19
0000171386 [app] INFO: Current step: 52
0000171726 [app] INFO: Current Resistance: 568.97, temperature 35.48, averaged temperature: 29.76
0000171734 [app] INFO: Input: 29.76, Setpoint: 30.00, Temp Diff: 0.24, Valve Output: 33.27
0000172664 [app] INFO: Current step: 51
0000172845 [app] INFO: Current Resistance: 570.64, temperature 36.35, averaged temperature: 31.91
0000172854 [app] INFO: Input: 31.91, Setpoint: 30.00, Temp Diff: -1.91, Valve Output: 22.41
0000173942 [app] INFO: Current step: 50
0000173947 [app] INFO: Stop valve...
0000173950 [app] INFO: Step difference to move valve: -28
0000173956 [app] INFO: Closing valve...
0000173966 [app] INFO: Current Resistance: 572.88, temperature 37.50, averaged temperature: 34.29
0000173974 [app] INFO: Input: 34.29, Setpoint: 30.00, Temp Diff: -4.29, Valve Output: 10.26
0000175087 [app] INFO: Current Resistance: 574.00, temperature 38.08, averaged temperature: 36.12
0000175095 [app] INFO: Input: 36.12, Setpoint: 30.00, Temp Diff: -6.12, Valve Output: 0.88
0000175234 [app] INFO: Current step: 49
0000176207 [app] INFO: Current Resistance: 574.56, temperature 38.37, averaged temperature: 37.16
0000176216 [app] INFO: Input: 37.16, Setpoint: 30.00, Temp Diff: -7.16, Valve Output: 0.00
0000176512 [app] INFO: Current step: 48
0000177328 [app] INFO: Current Resistance: 574.00, temperature 38.08, averaged temperature: 37.68
0000177337 [app] INFO: Input: 37.68, Setpoint: 30.00, Temp Diff: -7.68, Valve Output: 0.00
0000177790 [app] INFO: Current step: 47
0000178449 [app] INFO: Current Resistance: 574.56, temperature 38.37, averaged temperature: 38.08

```

Figur 24 Test temperature increase

```

0000800427 [app] INFO: Current Resistance: 572.32, temperature 37.21, averaged temperature: 37.62
0000800436 [app] INFO: Input: 37.62, Setpoint: 30.00, Temp Diff: -7.62, Valve Output: 0.00
0000801359 [app] INFO: Step difference to move valve: 0
0000801547 [app] INFO: Current Resistance: 568.97, temperature 35.48, averaged temperature: 37.16
0000801555 [app] INFO: Input: 37.16, Setpoint: 30.00, Temp Diff: -7.16, Valve Output: 1.02
0000802363 [app] INFO: Step difference to move valve: 1
0000802368 [app] INFO: Opening valve...
0000802667 [app] INFO: Current Resistance: 562.89, temperature 32.34, averaged temperature: 36.07
0000802674 [app] INFO: Input: 36.07, Setpoint: 30.00, Temp Diff: -6.07, Valve Output: 6.23
0000803646 [app] INFO: Current step: 1
0000803649 [app] INFO: Stop valve...
0000803653 [app] INFO: Step difference to move valve: 5
0000803657 [app] INFO: Opening valve...
0000803786 [app] INFO: Current Resistance: 560.69, temperature 31.20, averaged temperature: 34.75
0000803795 [app] INFO: Input: 34.75, Setpoint: 30.00, Temp Diff: -4.75, Valve Output: 12.61
0000804906 [app] INFO: Current Resistance: 557.96, temperature 29.79, averaged temperature: 33.20
0000804915 [app] INFO: Input: 33.20, Setpoint: 30.00, Temp Diff: -3.20, Valve Output: 20.18
0000804935 [app] INFO: Current step: 2
0000806027 [app] INFO: Current Resistance: 555.78, temperature 28.67, averaged temperature: 31.50
0000806035 [app] INFO: Input: 31.50, Setpoint: 30.00, Temp Diff: -1.50, Valve Output: 28.67
0000806213 [app] INFO: Current step: 3
0000807147 [app] INFO: Current Resistance: 553.07, temperature 27.27, averaged temperature: 29.85
0000807155 [app] INFO: Input: 29.85, Setpoint: 30.00, Temp Diff: 0.15, Valve Output: 36.89
0000807491 [app] INFO: Current step: 4
0000808266 [app] INFO: Current Resistance: 551.45, temperature 26.43, averaged temperature: 28.67
0000808275 [app] INFO: Input: 28.67, Setpoint: 30.00, Temp Diff: 1.33, Valve Output: 42.82
0000808769 [app] INFO: Current step: 5
0000809386 [app] INFO: Current Resistance: 549.83, temperature 25.60, averaged temperature: 27.55
0000809395 [app] INFO: Input: 27.55, Setpoint: 30.00, Temp Diff: 2.45, Valve Output: 48.54
0000810047 [app] INFO: Current step: 6
0000810050 [app] INFO: Stop valve...
0000810053 [app] INFO: Step difference to move valve: 42

```

Figur 25 Test temperature decrease


```
0000140133 [app] INFO: Current Temp: 26.60, Setpoint: 40.00, Temp Diff: 13.40, Valve Output: 100.00
0000140141 [app] INFO: Added data to log: timestamp: 1703018656, temperature: 26.60, valveOutput: 100.00
0000140169 [app] INFO: Requesting event (GetWeatherForecast) data
0000140187 [app] INFO: Step difference to move valve: 0
0000141177 [app] INFO: Event (GetWeatherForecast) data received
0000141183 [app] INFO: Weather forecast, currentTime: 1703018657, currentTemp 3.540000 C
0000141192 [app] INFO: Weather forecast, hour 1 Time: 1703016000, hour 1 Temp 3.760000 C
0000141199 [app] INFO: Weather forecast, hour 2 Time: 1703019600, hour 2 Temp 3.540000 C
0000141207 [app] INFO: Weather forecast, hour 3 Time: 1703023200, hour 3 Temp 3.760000 C
0000141192 [app] INFO: Step difference to move valve: 0
0000141319 [app] INFO: Current Resistance: 551.45, temperature 26.43, averaged temperature: 26.54
0000141327 [app] INFO: Input set to: 26.54
0000141330 [app] INFO: Current Temp: 26.54, Setpoint: 40.00, Temp Diff: 13.46, Valve Output: 100.00
0000141339 [app] INFO: Global setpoint set to: 30.00
0000142220 [app] INFO: Step difference to move valve: 0
0000142447 [app] INFO: Current Resistance: 551.99, temperature 26.71, averaged temperature: 26.60
0000142455 [app] INFO: Input set to: 26.60
0000142459 [app] INFO: Current Temp: 26.60, Setpoint: 30.00, Temp Diff: 3.40, Valve Output: 100.00
```

Figur 26 Test change setpoint based on weather forecast

Projekt konklusion

1. Projektets Resultater

- EH5IoT-projektet har opnået sit hovedmål om at optimere returtemperaturen i fjernvarmesystemer ved brug af IoT-teknologi.
- Integrationen af Particle Photon2, Kamstrup PT500 temperatursensoren og effektiv PID-logik har muliggjort præcis regulering af returtemperaturen til under 30°C.
- Anvendelsen af vejrprognosedata har forbedret temperaturreguleringen og bidraget til øget energieffektivitet og reducerede omkostninger.

2. Impact på Effektiviteten af Fjernvarmesystemet

- Systemet har forbedret energieffektiviteten signifikant ved at vedligeholde en lav returtemperatur, hvilket resulterer i en mere effektiv brug af termisk energi.
- En nedgang i energiforbruget og en reduktion af miljøpåvirkningen har understøttet bæredygtighedsindsatser og reduceret kulstofudledningen.

3. Udfordringer og Lærte Lektioner

- En væsentlig udfordring var at sikre effektiv og stabil kommunikation mellem systemkomponenterne, hvilket nødvendiggjorde finjustering af software og hardware.
- Grundig testning og tilpasningsevne har været afgørende i implementeringen af IoT-løsninger, idet projektet præsenterede kontinuerlige nye udfordringer og læringsmuligheder.

4. Forslag til Forbedringer og Fremtidige Udvidelser

- Vi vil fremover udforske mulighederne for at opdele applikationen i tråde for at forbedre respons og effektivitet, samt konvertering af eksisterende kode til C++ klasser for bedre struktur og vedligeholdelse.
- Energiopsparingsfunktioner og forbedret batteristatusindikation, som FuelGauge FuelGauge, vil blive integreret for at forlænge systemets levetid og pålidelighed.
- Implementering af en antifryse-funktion vil yde beskyttelse ved lave batteriniveauer, aktiveret kun når vejrudsigten viser negative temperaturer.
- Vi planlægger at anvende timer interrupts til forsinkede begivenheder, hvilket potentielt kan forbedre energisparetiltag.

- Yderligere vil vi beslutte præcist, hvilke data vi vil logge, herunder ventiltemperatur, nuværende udetemperatur, og ventilbevægelser, samt formatet og perioden for afsendelse til skyen for optimal overvågning og vedligeholdelse.

5. Generelle Konklusioner

- EH5IoT-projektet har vist, hvordan IoT-teknologier kan forbedre effektiviteten i centralvarmesystemer med direkte fordele som omkostningsbesparelser og bæredygtighed.
- Samarbejdet med industipartnere som Kredsløb A/S har været essentielt for projektets succes og har leveret et realistisk grundlag for implementering og testning.
- Disse konklusioner understreger projektets positive indvirkning på energieffektivitet og bæredygtighed og tilbyder en ramme for fremtidige forbedringer og udvidelser af systemet.



Referencer

- [1] Danfoss, »<https://ecl.portal.danfoss.dk/>,« [Online].
- [2] Particle, »Photon-2-datasheet,« 2023. [Online]. Available: <https://www.particle.io>.
- [3] T.-M. Valentin-Stefan, »Bilag mappe,« 2023.
- [4] irf520-power-mosfet-n-channel-module, »www.ardustore.dk,« 2023. [Online]. Available: <https://ardustore.dk/produkt/irf520-power-mosfet-n-channel-module>.
- [5] V.-S. Tudora-Mayer, »Github - EH5IoT-semester-project,« [Online]. Available: <https://github.com/ValentinSTM/EH5IoT-semester-project.git>.
- [6] Kamstrup, *Kamstrup A/S · Teknisk beskrivelse · 5512-2589_A2_DK_02.2021*, 2021.

Bilag

Hardware-dokumentation (skematiks, datasheets) i zip file.

Software kildekode.

<https://github.com/ValentinSTM/EH5IoT-semester-project.git>