

Titelblad med angivelse af projekttitel:

IoT Projekt 2

Navne:

Christian Theim Wulff Vexby, Daniel Thisted Larsen, Emil Senol,

Magnus Broder Møller, Oliver Phillip Friis

Gruppenummer: 4

Klasse: 2C

Opgave: IoT Projekt 2

Vejledere:

Bo Hansen, Dan Madsen, Kevin Lindemark Holm, Malene Hasse

Antal anslag:

61082

IoT Projekt 2

1. Resumé

I dette projekt bliver der undersøgt, hvorvidt indeklima har effekt på folkeskoleelevers evne til at fokusere samt modtage indlæring. Dette gøres blandt andet igennem virksomhedsfaget ved hjælp af projektledelses redskaber som Gantt, Scrum & WBS. Projektet har særligt fokus på at skabe en løsning til problemet som tager udgangspunkt i relevante faglige artikler om emnet.

Derudover vil der blive undersøgt ved hjælp af relevante faglige artikler, hvorvidt indeklima har en effekt på folkeskoleelevers evne til at koncentrere sig samt modtage indlæring. Særligt effekten af højt CO₂ i klasselokaler vil blive undersøgt.

Derefter vil der blive udarbejdet et IOT-produkt på baggrund af analyser i form af pathfinding, der via en sensor kan vise målinger, både på selve løsningen og med en graf på en hjemmeside.

Projektet konkluderer, at højt CO₂ niveau i klasselokalerne er et problem i folkeskolen og at vores IOT-løsning vil vise problemet. Grundet fejl i vores planlægning af tid, kom vi ikke igennem med vores anden prioritet som skulle have "opgraderet" hovedløsningen.

2. Indholdsfortegnelse

1. Resumé.....	3
2. Indholdsfortegnelse	4
3. Indledning.....	6
4. Problemformulering	7
Problemformulering:.....	7
Problemstilling:	7
Afgrænsning af problemformulering	7
5. Indledende undersøgelse.....	8
5.1 Ideudvikling	8
5.2 Research.....	8
5.3 Brugerundersøgelse	9
5.4 Persona og User stories.....	10
6. Kravspecifikation og accepttest metode.....	14
6.1 Rationale for prioritering af krav.....	18
7. Analyse	19
7.1 Arkitektur af pathfinding	19
7.1.1 Pathfinding af indlejrede systemer version 1	19
7.2.2 Pathfinding for programmering version 1.....	20
7.2 Den endelige arkitektur.....	21
7.2.1 Den endelige arkitektur for programmering.....	21
7.2.2 Den endelige arkitektur for indlejrede systemer	22
7.3 Teori af relevante elektroniske/programmering/netværk/server arkitekturblokke	23
7.3.1 Teori af relevante elektroniske arkitekturblokke.....	23
7.3.2 Programmering – Raspberry Pi 3 A+	27
7.3.3 Programmering – ESP32 Nr. 1	31
8. Løsningsdesign	34
8.1 Elektriske skematikker, målinger og udregninger (Indlejrede systemer)	34
8.2 Kode beskrivelse (Programmering)	35

9. Test af løsning	40
9.1 Pathfinding testresultater	40
9.2 Dokumentation af DUT	40
9.3 Udførelsel af Accepttest på DUT.....	42
10. Praktisk projektplanlægning og ledelse	50
10.1 WBS	50
10.2 Gantt.....	50
10.3 Scrum.....	52
10.4 Projektanalyse	53
10.4.1 Risikoanalyse	53
10.4.2 Interessentanalyse – IoT 2:	57
11. Konklusion	59
12. Projektforløbet	60
13. Perspektivering.....	61
14. Litteraturliste.....	62
Internet artikler:.....	62
15. Bilag	64
Bilag 1: Persona og userstories	64
Bilag 2: Business case	70
Bilag 3: Raspberry PI – log_MQ135.py	72
Bilag 4: Raspberry PI – MQ135_web.py	73
Bilag 5: ESP32 Nr.1 – boot.py	74
Bilag 6: ESP32 Nr.1 – main.py	75
Bilag 7: BOM Liste	77
Bilag 8: Spørgeskema – CO2 i klasselokaler.	78
Bilag 9: Skærm billeder af hjemmeside med 3 timers mellemrum.....	83

3. Indledning

FN's verdensmål 4.1.1, som har fokus på folkeskolebørns minimumskundskaber indenfor læsning og regning kobles op på indholdet og løsning af denne rapport.¹

Luftkvaliteten har stor betydning for vores velvære og påvirker os i alt hvad vi foretager os i dagligdagen. Det kan derfor være yderst problematisk, hvis der er for meget CO₂ i de lokaler man befinner sig i. For meget CO₂ påvirker både børn og voksnes koncentration og generelt velvære. I de danske folkeskoler er, der en stor tendens til, at være for meget CO₂ i lokalene. Dette kan have stor indflydelse på deres læring og forståelse af undervisningen og dermed have konsekvenser for deres videre færdens i livet.²

Denne rapport afdækker effekter, konsekvenser og information om hvordan CO₂ i større mængder påvirker mennesker og deres dagligdag. Uover information om CO₂ og dets påvirkning af mennesker kommer vi også med en mulig løsning af problemet ved hjælp af en IoT-løsning. Løsningen vil fremme opmærksomheden omkring CO₂ niveauet i et givent rum ved hjælp af visualisering. Dertil kommer vi med en løsningsforslag, der vil automatisere udluftningen i et rum, når CO₂ niveauet overstiger den givne grænse.

I rapporten ses hvordan vi har udarbejdet projektet i form af projektledelsesdiagrammer og -metoder. Der vil fremgå en dybdegående forklaring og begrundelse for valg af sensorer og komponenter til løsning af problemet. Fremvisning af både hardware- og softwareskematikker, med forklaring af kode og strømforbrug for komponenter og sensorer samt opsætning af kredsløb. Test af løsningens funktionalitet i praksis og overvejelser af mulige forbedringer til videreudvikling af løsningen i fremtiden.

¹ <https://www.verdensmaalene.dk/maal/4>

² <https://www.folkeskolen.dk/arbejdsliv-folkeskolen-nr-09-2019-fysisk-arbejdsmiljo/du-bliver-dummere-i-darlig-luft/1375839>

4. Problemformulering

Problemformulering:

- Hvad kan vi gøre for at skabe et bedre indeklima i folkeskoler ved hjælp af en IoT løsning.

Problemstilling:

- Hvad er de primære faktorer som kan føre til dårligt indeklima?

Problemstilling:

- Hvilke konsekvenser har dårligt indeklima på sundhed og trivsel i undervisningen?

Problemstilling:

- Hvordan kan man ved hjælp af sensorer måle de forhold, der skaber et dårligt indeklima?

Problemstilling:

- Hvilke relevante data kan sensoren samle til databasen og have en grafisk fremstilling?

Afgrænsning af problemformulering

I denne rapport har vi valgt at afgrænse vores projekt til grundskoler. Dette er gjort på baggrund af det valgte verdensmål (4.1.1), som fokuserer på minimumsfærdigheder i henholdsvis slutningen af 5. klasse og 9.klasse i regning og læsning. I og med at vores verdensmål fokuserer på grundskole, ser vi ikke grund til at finde relevant data, og problemløse på gymnasial og universitet niveau. Derudover har vi valgt at fokusere på hvordan CO₂ har en effekt på indeklima, fremfor at fokusere på flere faktorer der kan være med til at skabe et dårligt indeklima. Vi har valgt at fokusere på verdensmål 4.1.1, med fokus på CO₂ i klasselokalerne, på baggrund af den mængde PPM (parts per million) der er i folkeskole klasselokaler kontra gymnasie/universitets lokaler.³

³https://www.exhausto.dk/~media/Global/PDF/Institute/Research/Officiel%20Rapport%20resultatrapport_mx2014_5_small.pdf

5. Indledende undersøgelse

5.1 Ideudvikling

Ved ideudvikling begyndte vi med at anvende en brainstorm teknik, der hedder negativ brainstorm. Negativ brainstorm handler om at komme i gang med den kreative proces. Vi anvendte den til at komme i gang, ved at komme på så mange dårlige idéer vi kunne. Dette hjalp os til at tænke kreativt og ud af boksen.

Det var efter den negative brainstorm, at et gruppemedlem kom i tanke om et podcast han havde lyttet til. Podcastet handlede om faren ved CO₂ og hvordan CO₂ påvirker hjernen. Efter langvarige overvejelser, kom vi frem til at vi ville fokusere på folkeskoler, da vi fandt ud af at der i de fleste folkeskoleklasselokaler er en for høj mængde CO₂. Det har stor betydning for deres læring, at der er et sundt indeklima så de kan koncentrere sig i undervisningen.

I forbindelse med at vi fandt ud af at der typisk er et for højt CO₂ niveau i folkeskolernes klasselokaler, kom vi frem til at verdensmål 4.1.1 passede perfekt til det problem vi undersøgte⁴.

5.2 Research

Dårligt indeklima er et problem ude i klasselokalerne. Det har en stor effekt på børns indlæring. Børn tilbringer omkring 20% af deres tid på skolen, når de ikke sover. Derfor er det vigtigt, at klasselokalerne er sunde at være i.

Udenfor er der typisk cirka 400 ppm CO₂ i luften. Det svinger lidt, specielt fra by til by. Indenfor må der være op til 1000pm, før det begynder at være til gene. Her begynder man at blive sløv og trække vejret lidt værre. Mellem 2000-5000 ppm CO₂ i luften begynder koncentrationen at blive dårlig, hjertet banker hurtigere, hovedpine og kvalme kan forekomme. Det er ikke unormalt, at et klasselokale kommer til at ligge over de 2000 ppm, men vil typisk ikke komme til næste skadelige trin.

5

Når CO₂'en i luften bliver høj tyder det også på andre problemer, så som virus, støv, skimmelsvamp og/eller meldug. Derfor er det en god ide for skoler at måle CO₂ i luften, da det er en nem metode til at finde ud af luftkvaliteten. Dårligt indeklima kan føre til sygdom, irritation i næse, øjne eller halsen

⁴https://www.exhausto.dk/~media/Global/PDF/Institute/Research/Officiel%20Rapport%20resultatrappor_m_x2014_5_small.pdf

⁵ <https://www.co2meter.com/blogs/news/indoor-air-quality-monitoring-schools-classrooms>

Den simpleste måde at forbedre ved brug af udluftning. Som skrevet tidligere, så har luften udenfor en mindre koncentration af CO₂, så et åbent vindue vil udligne CO₂ niveauet, og udskifte luften inde i klasselokalet.

Vores løsning viser problemer med CO₂, men det er ikke altid at åbne vinduer er nok til at få niveauet helt ned på et sundt niveau. I England har de mange CO₂ målere i brug, men har ikke god nok ventilation til at forbedre problemet på alle skoler. Åbne vinduer gør at temperaturen også falder. HSE (Health and Safety Executive), anbefaler, at temperaturen ikke må være mindre end 16 grader i skoler og arbejdspladser. Normal stuetemperatur vil være omkring 21 grader og det er, hvor de fleste vil trives. Når måleren siger, der er for meget CO₂ i luften, og vinduerne er åbne, så kræver det, at ventilationen er bedre i bygningen. Åbne vinduer øger også varmeregningen. En skole i Survey har set en øgning på 40.000£, altså omkring 340.000 kr. på grund af åbne vinduer efter at have installeret CO₂ målere.⁶

Ifølge en artikel fra 2018 har Københavns kommune haft et fokus på at skabe et bedre indeklima i skoler og institutioner. Det har de gjort for eksempel med IOT-sensorer lignende vores produkt.

Mange andre kommuner i Danmark har også forsøgt med IOT-sensorer for at nedbringe CO₂ niveauet i klasselokalerne.⁷ Her har Furesø kommune kørt et forsøg i slut 2021. DTU kører også lignende eksperimenter på mange skoler i Danmark, hvor de har fundet ud af, at niveauet af CO₂ ligger for højt i over 90% af klasselokalerne.⁸

5.3 Brugerundersøgelse

Vores brugerundersøgelse i projektet består af et spørgeskema, som vi har lavet på surveymonkey.com, til lærerne på skolen ved sundet. Spørgeskemaet er lavet til at få en forståelse for hvilke overvejelser folkeskolelærere laver omkring dårligt indeklima, og om hvorvidt de gør noget ved det. Klassetrinene, der er repræsenteret, er folkeskoleklasser fra 0-9 klasse. Undersøgelsen viste at det ville blive meget nemmere for folkeskolelærerne at automatisere udluftningen. Størstedelen af lærerne synes også det ville være en hjælp for dem, hvis der var en visuel indikator, når indeklimaet blev usundt, så de præcis ved hvornår der skal luftes ud i lokalet. Der var stor diversitet i forhold til om de vidste hvad konsekvenserne ved for meget CO₂ i et klasselokale indebar og det samme viste

⁶ <https://schoolsweek.co.uk/classroom-carbon-dioxide-levels-three-times-above-watchdog-guidelines/>

⁷ <https://cphsolutionslab.dk/news/forbedring-af-indeklimaet-i-skoler-og-daginstitutioner>

⁸ <https://www.folkeskolen.dk/arbejdsliv-borneliv-fureso-kommune/masseeksperimentet-co2-niveauet-i-klassen-spraenger-skalaen/1377365>

sig ved om de var klar over, at klasselokaler oftest var overbelastet i mængden af CO₂. Ud fra besvarelserne på brugerundersøgelsen kan vi se, at det er et problem med indeklima flere steder på folkeskolen. Det ville gavne lærerne at automatisere udluftning i klasserne og en visuel indikator ville være behjælpelig for lærerne.⁹

Vi har valgt at udspørge lærere på baggrund af deres indflydelse på deres arbejdsplads, og deres mulighed for at forbedre indeklimaet i deres klasselokaler.

5.4 Persona og User stories

Persona'er til projektet er personer, der enten arbejder på en skole eller er elev.

Navn: Oskar Nielsen.



Figur 1 fra: <https://www.colourbox.dk/billeder/mand-office-pc-billeder-56900959>

- **Alder:**
 - 55
- **Job:**
 - Rektor
- **Påklædning:**
 - Skjorte og åben habit med habit bukser.
- **Personlige og professionelle ambitioner:**
 - Drømmer om at kunne blive rektor på et universitet som betaler bedre, og at kunne tage sig af sin familie også efter han er væk økonomisk.

⁹ Se bilag 8 for at se spørgeskemaet og dets besvarelse

- **Problemer og udfordringer:**
 - Har svært ved at finde personale der vælger at blive på skolen i længere tid
- **Interesser/Hobbyer:**
 - Går til senior fodbold 2 gange om ugen, og er stor Brøndby fan.
- **Adfærd på sociale medier:**
 - Anvender kun Facebook til at poste billeder af sin familie når de tager på familieture.
Bliver dog nogle gange fanget af nyhedsmedier på Facebook.
- **IT-kompetencer:**
 - Har et kursus i Excel, og bruger sin computer til administrativt arbejde hver dag.

User story:

Som rektor tilbringer jeg mycket af min tid på skolen og skal sørge for, at mine lærere og elever har den bedste forudsætning for en god skoledag. Specielt indeklimaet har et stort fokus nu.	Story nr. 1
Jeg ønsker en løsning, der kan hjælpe med at holde et godt indeklima	Prioritet: 5
Da mit mål er at sørge for, at skolen er så effektiv som muligt er et godt indeklima vigtigt.	Effort: 5

Personaen Oskar er den person, der tænker mest over at forbedre hverdagen for hele skolen, både elever og lærere. Det er hans job at sørge for at skolen fungerer så godt som muligt, derfor er hans ambition meget høj og prioriterer det også højt. Oskar har også det problem, at han har svært ved at bideholde lærere. Jo bedre forhold der er på skolen, jo større chance er der for at lærere har lyst til at blive på skolen. Her er bedre luftkvalitet en af de ting, der kan forbedres. Hvis man altid er træt på arbejdet, så har man ikke lyst til at beholde det i lang tid og vil hurtigt lede efter nyt arbejde.

Oskar er relevant at få overbevist om at få denne idé implementeret, da han har en indflydelse på hvad der sker på skolen, og han kan skrive forslag til kommunen, som vil tage det i deres overvejelser.

Navn: Alex Nepalsen*Figur 2 fra: <https://www.colourbox.dk/billeder/teenager-teen-latino-billeder-56784657>*

- **Alder:**
 - 15
- **Job:**
 - Salgsassistent deltid, Teknikafdelingen i Bilka. Elevråds repræsentant.
- **Påklædning:**
 - nørdet/casual
- **Personlige og professionelle ambitioner:**
 - Vil gerne have sit eget firma en dag hvor han kan tjene penge.
- **Problemer og udfordringer:**
 - Vil gerne være mere social men har svært ved at balancere det sociale og uddannelse.
- **Interesser/Hobbyer:**
 - IT, rejser & kultur.
- **Adfærd på sociale medier:**
 - Følger IT sider og deltager ofte i diskussioner hvor han bringer hans meninger og ideer ind.
- **IT-kompetencer:**
 - Office pakken, Python & en del forskelligt hardware.

User story:

I og med at jeg arbejder med Tech support, så bruger jeg meget tid indenfor, og foran en computer. Når der er gået tilstrækkelig lang tid, begynder luften og blive tung og jeg kan mærke at koncentrationen den falder lige så langsomt.	Story nr. 4
På arbejdet er det vigtigt at være frisk og klar til at hjælpe dem der har brug for hjælp. Derfor er det vigtigt for mig at selv i de sene timer på arbejdet, at jeg er koncentreret.	Prioritet: 3
Jeg syntes at godt indeklima er nemt at opnå, i og med at der kun skal åbnes et vindue eller en dør. Det kan dog være rart at vide lige præcis hvornår det ville være bedst at åbne et vindue, specielt om vinteren hvor man ikke har lyst til at have vinduet åbent hele tiden.	Effort: 2

Alex, der kan lide IT, er en anden Persona, der kunne finde vores produkt interessant. Da han arbejder i elektronikafdelingen i Bilka, så følger han også med i teknologi nyheder og diverse gadgets. Det er her vi kan fange ham og få ham med på ideen om at få vores IOT-produkt til hans klasseværelse og skole. Som elevrådsrepræsentant har han også en tættere forbindelse til skolen end andre elever og har indflydelse på skolen.

6. Kravspecifikation og accepttest metode

ID: 1	Krav: Løsningen skal være robust.	Prioritet: 1
Kategori: HW	<p>Accept test:</p> <p>Step 1) Løsningen tændes.</p> <p>Step 2) Løsningen rystes i 10 sekunder.</p> <p>Hvis løsningen ikke har taget skade og man ikke kan høre noget inde fra kassen, vurderes kravet som værende bestået.</p>	Passed/ Failed/ Ikke testet

ID: 2	Krav: Løsningen skal kunne give en visuel markør, når CO2 niveauet overstiger den anbefalet mængde i et rum.	Prioritet: 1
Kategori: Visuelt	<p>Accept test:</p> <p>Step 1) Løsningen tændes.</p> <p>Step 2) Der åndes på sensoren indtil den visuelle indikator skifter farve til rød.</p> <p>Hvis måleren kommer op over 800 PPM og at den visuelle markør bliver skiftet til farven rød, vurderes kravet som værende bestået.</p>	Passed/ Failed/ Ikke testet

ID: 3	Krav: Løsningen skal kunne forbindes til et lokalnetværk.	Prioritet: 1
Kategori: Forbindelse	<p>Accept test:</p> <p>Step 1) Løsningen tilsættes strøm forsyning.</p> <p>Step 2) Efter 1 minut skal løsningen være tilkoblet lokalnetværket.</p> <p>Step 3) Løsningen pinges fra en computer.</p> <p>Hvis løsningen har forbindelse til lokalnetværket efter 1 minut og ingen ping-pakker går tabt, vurderes kravet som bestået.</p>	Passed/ Failed/ Ikke testet

ID: 4	Krav: Kunne vise en graf over CO2 niveau på en hjemmeside.	Prioritet: 1
Kategori: Programmering	<p>Accept test:</p> <p>Step 1) Løsningen tændes.</p> <p>Step 2) En computer opretter forbindelse til løsningens hjemmeside.</p> <p>Step 3) Der kontrolleres om hjemmesiden viser en graf der indeholder data fra CO2 måleren.</p> <p>Tallene fra Shell sammenlignes med tallene fra hjemmesiden og tjekker om den viser en passende graf over CO2 PPM, hvis tallene passer, vurderes kravet som bestået.</p>	Passed/ Failed/ Ikke testet

ID: 5	Krav: Grafen på hjemmesiden skal kunne opdateres hvert 5 minut med ny data fra de sidste 5 minutter	Prioritet: 1
Kategori: Prog	<p>Accept test:</p> <p>Step 1) Løsningen tændes.</p> <p>Step 2) Grafen opdateres ved et tryk på refresh</p> <p>Step 3) Grafen opdateres og tjekkes hvert 5 minutter for opdateringer over CO2 niveauet i lokalet.</p> <p>Hvis der er ny data hvert. 5 minut vurderes kravet som bestået.</p>	Passed/ Failed/ Ikke testet:

ID: 6	Krav: Løsningen skal have dokumentation på at den er testet i minimum 3 timer	Prioritet: 1
Kategori: Power	<p>Accept test:</p> <p>Step 1) Løsningen bliver sat op, og tilsluttes strømkilde.</p> <p>Step 2) Timer startes på 3 timer</p> <p>Når ingen problemer opstår efter 3 times brug, vurderes kravet som bestået.</p>	Passed/ Failed/ Ikke testet:

ID: 7	Krav: Løsningen skal kunne forbinde trådløst til en mikrocontroller.	Prioritet: 2
Kategori: Forbindelse	<p>Accept test:</p> <p>Step 1) Løsningen & mikrocontroller tændes</p> <p>Step 2) Løsningen sender besked til mikrocontroller trådløst</p> <p>Hvis besked bliver verificeret som det der blev afsendt, vurderes kravet som bestået.</p>	Passed/ Failed/ Ikke testet:

ID: 8	Krav: Løsningen skal være i stand til at åbne et vindue, ved for højt CO2 niveau i rummet.	Prioritet: 2
Kategori: HW/Prog	<p>Accept test:</p> <p>Step 1) Løsningen tændes</p> <p>Step 2) Der åndes på måleren til ppma når >800</p> <p>Hvis måleren sender signal og åbner vinduet ved hjælp af motoren, vurderes kravet som bestået.</p>	Passed/ Failed/ Ikke testet:

6.1 Rationale for prioritering af krav

ID: 1

Vi har valgt at give dette krav prioritet 1 på grund af målgruppen. Der kan blive spillet bold, kastede med ting og hoppet rundt i et folkeskoleklasseværelse hvilket alt sammen kan være årsag til løsningen nemt kunne blive beskadiget.

ID: 2

Vi har valgt at give dette krav prioritet 1 eftersom at det ville hjælpe løsningens kvalitet at have et visuelt aspekt. Det hjælper på brugeroplevelsen og gør løsningen nemmere at anvende korrekt.

ID: 3

Vi har valgt at give dette krav prioritet 1 siden at det skaber en linje imellem de forskellige teknologier i løsningen hvor data kan transporteres. Dette hjælper også generelt med lagring af data online.

ID: 4

Vi har valgt at give dette krav prioritet 1 fordi kravet vil hjælpe brugeren med at komme længere ned i detaljerne, men samtidig gøre det mere visuelt.

ID: 5

Vi har valgt at give dette krav prioritet 1 på grund af at et engangsbillede ikke giver nok information til at gøre vores løsning funktionel og give brugeren den kontinuerlige mængde af data der ønskes.

ID: 6

Vi har valgt at give dette krav prioritet 1 eftersom at det hjælper på funktionaliteten af løsningen og dens evne til at udføre dens arbejde bedst muligt.

ID: 7

Vi har valgt at give dette krav prioritet 2 siden at kravet var tiltænkt at være en "opgradering" på den originale ide, hvilket ville have været implementeret, hvis der havde været tid til overs.

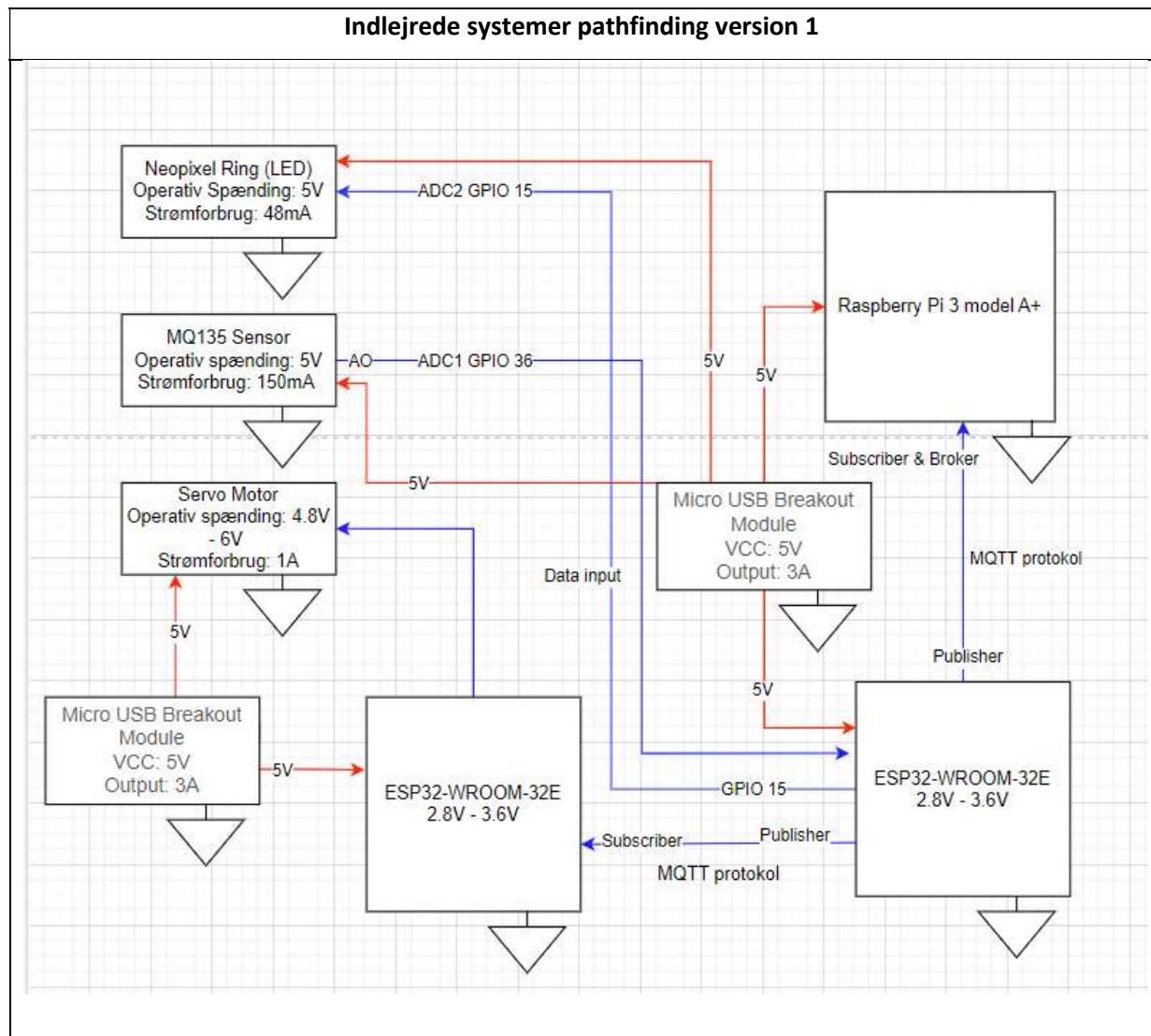
ID: 8

Vi har valgt at give dette krav prioritet 2 siden at det var en forlængelse af ID: 7 som kunne have været implementeret hvis der havde været tid til overs.

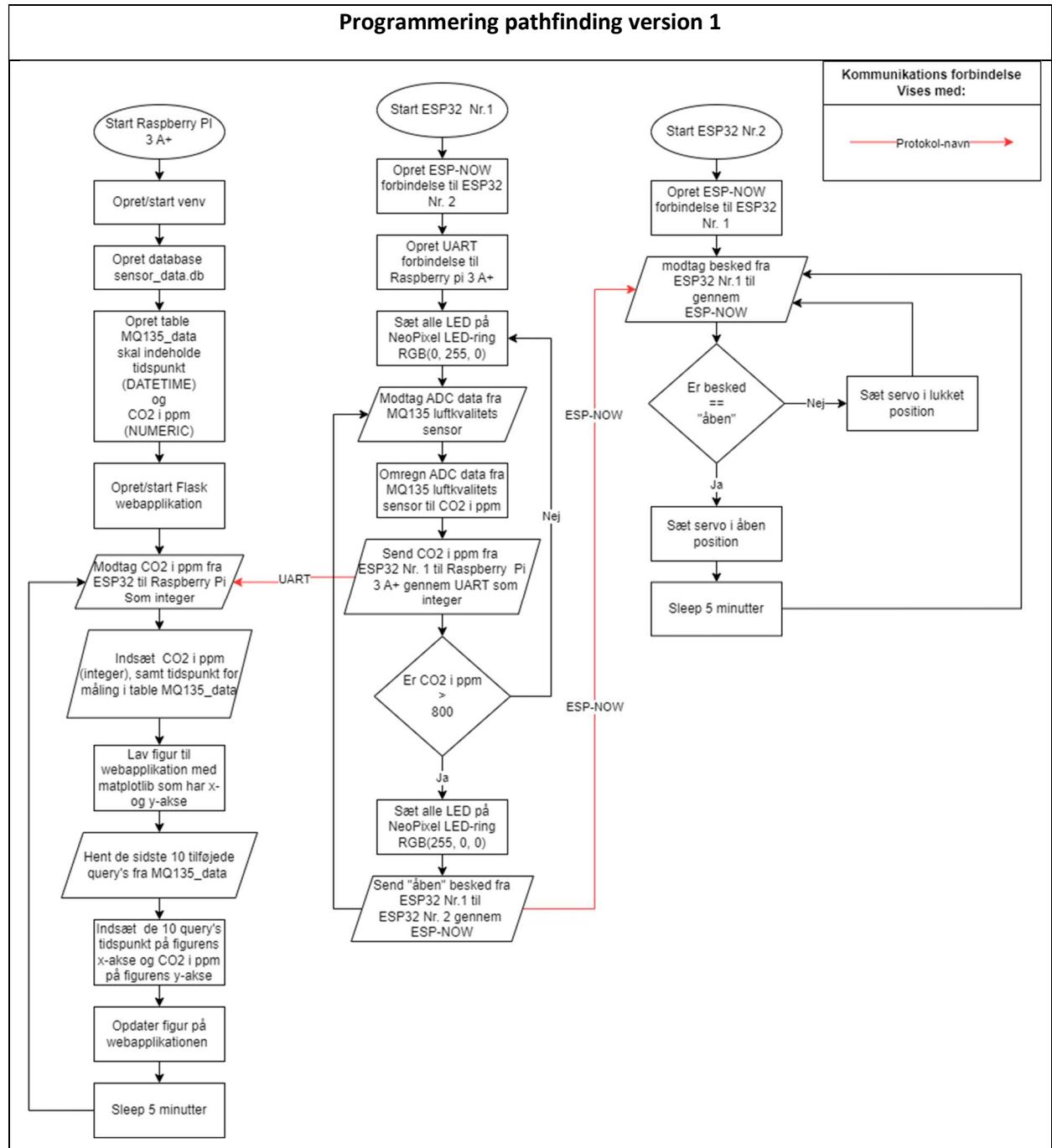
7. Analyse

7.1 Arkitektur af pathfinding

7.1.1 Pathfinding af indlejrede systemer version 1.

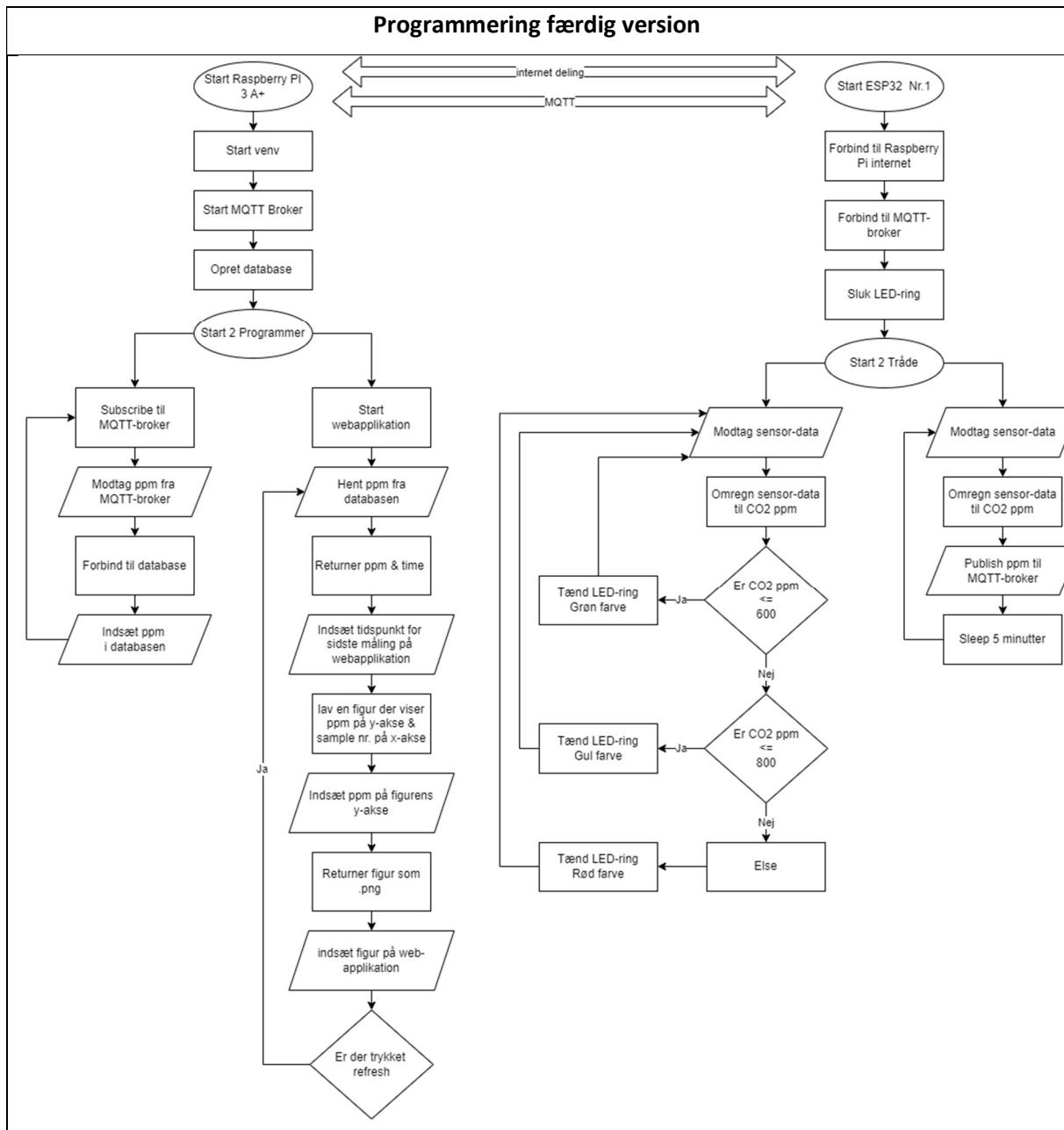


7.2.2 Pathfinding for programmering version 1.

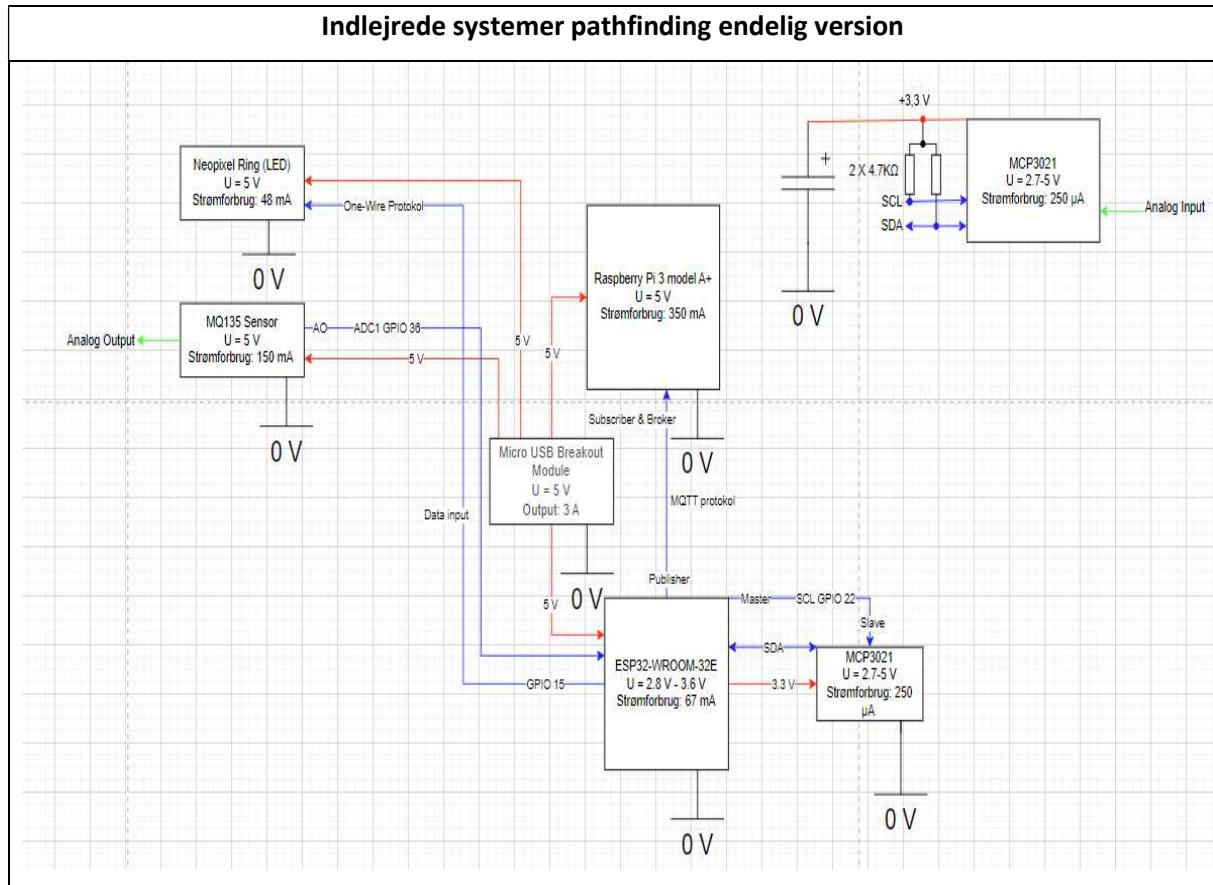


7.2 Den endelige arkitektur

7.2.1 Den endelige arkitektur for programmering.



7.2.2 Den endelige arkitektur for indlejrede systemer



7.3 Teori af relevante elektroniske/programmering/netværk/server arkitekturblokke

7.3.1 Teori af relevante elektroniske arkitekturblokke.

Blok navn: Neopixel LED ring	
	<p>Neopixel ringen bruges til at lyse forskellige farver ved hjælp af RGB efter eget valg. Det gør det til en god visuel indikator.</p> <p>Den har en ydre diameter på 37mm, indre diameter på 23 mm og en tykkelse på 7mm.</p> <p>Den har en Operativ spænding på 5V og et strømforbrug på 20mA for hvert farve pixel. Den består af 12 LED'er, der hver kan programmes til at vise en speciel farve.</p> <p>I vores projekt bliver Neopixel LED ringen brugt til at indikere CO₂ niveauet i rummet. Den lyser grønt ved et sundt indeklima og rødt ved et usundt indeklima.</p> <p>NeoPixel ringen bruger One-Wire protokollen.</p>

Blok navn: ESP32-WROOM-32E



ESP32 er en mikrocontroller også kaldet en minicomputer, den består af det en almindelig computer indeholder såsom, RAM, CPU, og har input og outputs i form af pins.

Den har en forsyningsspænding på 5V og en IO (pins) spænding på 3.3V.

Dens flash hukommelse er på 4MB og clockfrekvens på 240 MHz.

Dens strømforbrug er 67mA og strømforbruget ved dvale er 350uA.

I projektet er ESP32 tilkoblet en MQ135, som mäter CO₂ i luften og sender det tilbage til ESP32'en. ESP32 sender denne information videre til vores Raspberry pi ved hjælp af MQTT-protokollen, herefter laver Raspberry pi en graf over CO₂ niveauet.

ESP32 anvender også 2 hardware protokoller, I2C til vores MCP3021 og One Wire til vores NeoPixel.

Raspberry Pi 3 Model A+



Raspberry Pi 3 Model A+ er en minicomputer, der kører på Linux operativsystemet Raspberry Pi OS. Den består af det samme, som en "normal" computer bare som fx ram og CPU. Udover at være en computer, har en Raspberry pi også pins som man kan bruge til input og output.

Raspberry pi 3A+ har en inputspænding på 5V via USB og 5V DC via GPIO header.

Raspberry pi bliver brugt til at hoste webserver, indeholder en database og bliver brugt som MQTT-broker.

MQ135 luftkvalitet sensor

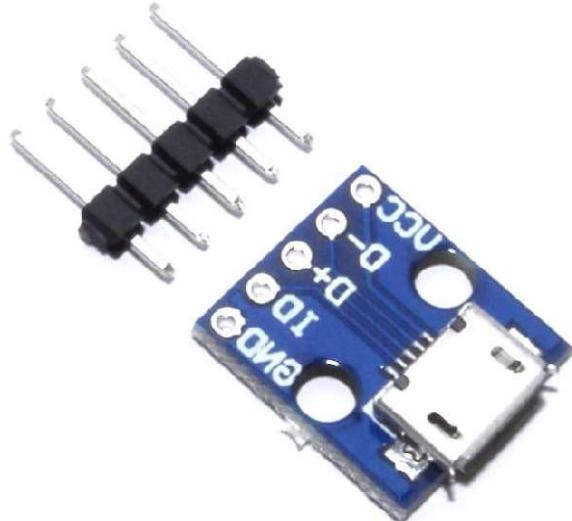


MQ135 er en sensor, der bruges til at måle luftkvaliteten på forskellige parameter. Den kan måle mængden af ammoniak, sulfider, alkohol, benzen, røg og CO₂ i et lokale.

Den har en Operativ spænding på 5V og et strømförbrug på 150mA.

I vores projekt bliver den brugt til at måle luftkvaliteten i et folkeskoleklasselokale med særligt fokus på mængden af CO₂ i luften.

Micro USB Breakout module

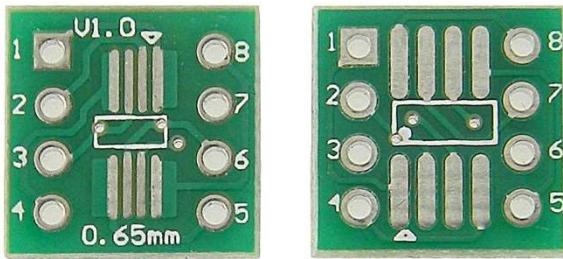


I projektet bruger vi et Micro USB Breakout Module, der bliver forsynt med 5V gennem et USB-kabel. Udeover USB-kablet har den også 5 pins.

I projektet bliver vores Breakout module brugt til at forsyne vores NeoPixel, MQ135, ESP32 og Raspberry pi 3A+ med 5V ved hjælp af USB.

Den har en på Længde 14mm en bredde på 13mm en højde på 4,2mm og en vægt på 2g.

MCP3021	
 A black MCP3021 integrated circuit in a surface-mount package, showing its pins and lead frame.	<p>MCP3021 er en A/D konverter også kaldet en ADC med en 10-bit resolution.</p> <p>Den har en maks. konverteringsstrøm på $250\mu\text{A}$ og en idle strøm på $1 \mu\text{A}$.</p> <p>I projektet bruger MCP3021 protokollen I2C.</p> <p>Vi bruger MCP3021 i forbindelse med et 8-pin breakout module, som vi har loddet den på.</p>

8-Pin breakout module	
 Two photographs of an 8-pin breakout module. The left image shows the top side with pads labeled 1 through 8 and a dimension of "0.65mm" between pins 4 and 5. The right image shows the bottom side with the same pin layout and a central rectangular component.	<p>8-Pin breakout module bliver i projektet brugt sammen med en MCP3021(ADC), som er blevet loddet på i midten, så vi kan tilgå pins på den.</p>

7.3.2 Programmering – Raspberry Pi 3 A+

Blok navn: Start venv

Starter et virtual-environment(venv). Det gør det muligt at installere Python pakker til vores programmer, uden de har en indflydelse på hvad der ellers kører på vores RPI(Raspberry Pi). For at oprette et venv på Linux bruger man kommandoen:

```
$ python3 -m venv IoT2
```

Vi har kaldt vores venv IoT2. For at starte vores venv går vi ind i IoT2 mappen og skriver kommandoen:

```
$ source ./bin/activate
```

For at deaktivere ens .venv bruges kommandoen:

```
$ deactivate
```

Blok navn: Start MQTT-broker

Vi bruger MQTT protokollen til at sende data fra vores ESP32 til vores RPI. Til det skal vi bruge en MQTT-broker på vores RPI. Vi har installeret MQTT-brokeren Mosquitto ved at bruge kommandoen:

```
$ sudo apt install mosquitto mosquitto-clients
```

Herefter skal man bevæge sig til /etc/mosquitto åbne .conf filen og tilføje linjerne:

Listener 1883 0.0.0.0

Allow_anonymous true

De to linjer der bliver tilføjet .conf filen tillader andre enheder at forbinde til Mosquitto MQTT-broker.

Blok navn: Opret database

Til at oprette en database bruger vi SQLite3 da det allerede kommer med python. For at installere SQLite3 på vores RPI, skriver man:

```
$ sudo apt inst  
all sqlite3
```

Så går vi ind i den mappe hvor vi vil have vores database og skriver:

```
$ sqlite3 sensor_data.db
```

Herefter opretter vi et table ved at skrive:

```
sqlite> CREATE TABLE MQ135_data (timestamp DATETIME, ppm NUMERIC);
```

Det laver et table der kan indeholde tidspunkt og ppm.

Blok navn: Start 2 programmer

Her starter vi to programmer, det ene har vi kaldt log_MQ135.py som logger sensor data til databasen og det andet er MQ135_web.py som er vores flask applikation.

Først starter vi log_MQ135.py ved at skrive:

```
$ nohup python3 log_MQ135.py &
```

nohup gør at programmet ikke stopper selv når vi stopper vores ssh forbindelse. & tegnet er vigtigt at skrive til sidst, det gør at programmet kører i baggrunden. Det gør også at vi kan køre flere programmer på en gang.

For at starte vores flask applikation skriver vi:

```
$ flask --app MQ135_web run --debug --host=0.0.0.0
```

--debug gør at vi kan redigere i programmet og bare genindlæse hjemmesiden for at se ændringer.
--host=0.0.0.0 gør at andre på netværket kan tilgå vores hjemmeside.

Blok navn: Subscribe til MQTT-broker

I vores log_MQ135.py skal vi subscribe til vores MQTT-broker, dette gør vi ved at bruge nogle funktioner fra et python modul kaldet paho-mqtt. Det installeres med pip inde i vores venv ved at skrive.

```
$ pip install paho-mqtt
```

Vi bruger subscribe() funktionen fra dette modul til at tilslutte vores MQTT-broker.

Blok navn: Modtag ppm fra MQTT-broker

Når nu vi har subscibet til MQTT-brokeren, modtager programmet data.

Blok navn: Forbind til database

Begge programmer på vores RPI forbinder til vores SQLite3 database. derefter kan vi bruge sqlite3.connect('sti-navn/database-navn.db) man kan også give funktionen et argument mere som gør det muligt at tilgå databaser der ikke kører i samme tråd. Dette gøres ved at tilføje argumentet: check_same_thread=False

Blok navn: Indsæt ppm i databasen

For at indsætte ppm vi har fået fra vores MQTT-broker på vores table i sensor_data.db har vi lavet en funktion kaldet log_data(). Den får ppm som argument, laver en forbindelse til databasen og indsætter tidspunkt og ppm i databasen.

Blok navn: Start webapplikation

Vores webapplikation bruger Flask som er lightweight og derfor ikke tager for mange ressourcer fra vores RPI. Dog er det ikke anbefalet at bruge Flask til hjemmesider der kommer ud fra lokalnetværket. For at starte vores Flask webapplikation bruger vi kommandoen:

```
$ flask -app MQ135_web run --debug --host=0.0.0.0
```

Blok navn: Hent ppm fra databasen

For at hente ppm fra databasen har vi lavet en funktion kaldet `get_hist_data()`. Denne funktion gør brug af `fetchall()` funktionen fra cursor i sqlite3 modulet. Derfor er sqlite3 blevet importeret. `fetchall()` tager alle rows af en query og returnere en liste med ppm og date.

Blok navn: Indsæt nyeste data på webapplikation

For at indsætte nyeste data på webapplikation har vi lavet en funktion kaldet `get_last_data()`. Denne funktion gør brug af `cursor()` og `execute()` funktionen. Derfor er sqlite3 blevet importeret. I `execute()` funktionen skriver vi "SELECT * FROM MQ135_data ORDER BY timestamp DESC LIMIT 1". Det gør at vi tager alt fra vores table og sortere det efter timestamp i aftagende orden og tager kun den første. Herefter returnerer funktionen time og ppm. De bliver så sat ind i vores index.html fil der ligger under templates ved at skrive `{{time}}` og `{{ppm}}`.

Blok navn: Lav figur

For at lave en figur/graf til webapplikationen bruger vi en python pakke kaldet matplotlib, den installeres i vores venv ved at skrive:

```
$ pip install matplotlib
```

Og så skal det importeres i vores MQ135_web.py program.

Blok navn: Indsæt ppm på figurens y-akse

For at indsætte ppm på figurens y-akse bruger vi `Figure()` klassen fra matplotlib som er blevet importeret i vores python script. Der bliver lavet en instans af klassen i en funktion vi har kaldet `plot_ppm()` funktionen bruger `get_hist_data()` til at hente dataen der skal sættes i på figuren.

Blok navn: Indsæt figur på webapplikation

For at indsætte figuren på webapplikationen `@app.route('/plot/ppm')` til at specificere at denne funktion skal køres når en klient sender en GET-request til /plot/ppm URL'en.

Blok navn: Er der trykket refresh

Hvis der er trykket refresh vil den nye data hvis der er noget, bliver indsat på hjemmesiden.

7.3.3 Programmering – ESP32 Nr. 1

Blok navn: Forbind til Raspberry Pi internet

For at kunne forbinde til det samme netværk som vores RPI, skal vi bruge micropython modulet network på vores ESP32. Vi laver en funktion i vores boot.py vi kalder do_connect(), vores funktion kontrollere om der er etableret internetforbindelse. Hvis ikke der er forsøger den at forbinde til et netværk med funktionen .connect('SSID', 'PASSPHRASE'). Inden vi kører do_connect bruger vi sleep(60) funktionen fra time modulet, derfor har vi importeret sleep fra time. Det har vi gjort for at give vores RPI mulighed for at boote op inden vores ESP32.

Blok navn: Forbind til MQTT-broker

For at kunne forbinde til vores MQTT-broker skal vi bruge MQTTClient() klassen i vores main.py på ESP32. Til det har vi downloadet umqtt.simple fra et github repository.¹⁰ Vi bruger .connect() funktionen fra MQTTclient til at oprette forbindelse til brokeren.

Blok navn: Sluk LED-ring

For at kunne slukke vores Neopixel LED-ring har vi lavet en funktion vi kalder np_off() hvori vi bruger NeoPixel() klassen fra neopixel modulet som skal importeres til vores main.py. I funktionen laver vi et for-loop der kører 12 gange (antallet af LED'er på neopixel LED-ringen) giver LED'en rgb-værdien (0, 0, 0) og bruger .write metoden til at skrive rgb værdien til neopixel LED-ringen.

Blok navn: Start 2 Tråde

For at starte 2 tråde i vores main.py på ESP32, gør vi brug af funktionen .start_new_thread(funktionens_navn,()). Funktionen ligger i _thread modulet som bliver importeret i vores main.py script. Vi bruger funktionen to gange for at starte tråde

¹⁰ <https://github.com/micropython/micropython-lib/tree/master/micropython/umqtt.simple>

Blok navn: Modtag sensor-data

For at modtage sensor data fra vores MCP3021 adc som er forbundet til MQ135 sensoren, bruger vi I2C. Til det importere vi I2C Klassen og Pin klassen fra machine modulet. Vi laver en instans af klassen I2C(scl=Pin(22), sda=Pin(21), freq=400000) som vi kan bruge i en funktion vi kalder read_adc(). Her bruger vi .readfrom_into() funktionen til at læse dataen vi modtager fra i2c forbindelsen.

Blok navn: Omregn sensor-data til CO2 ppm

For at omregne sensor data til co2 i ppm har vi lavet en funktion vi kalder adc_to_ppm(). Funktionen kortlægger adc målingerne til CO2 koncentration og returnere ppm.

Blok navn: Er CO2 ppm <= 600

For at kontrollere om CO2 i ppm er mindre eller lig 600 bruger vi et if-statement som kontrolstruktur. Hvis er mindre eller lig 600 går forsætter programmet til "Tænd LED-ring Grøn farve" blokken.

Blok navn: Tænd LED-ring Grøn farve

For at tænde neopixel led ringen i grøn har vi fulgt samme princip som "Sluk LED-ring" blokken. Her har vi bare lavet en funktion vi kalder np_green() som sender rgb værdien (0, 50, 0) til neopixel LED-ringen.

Blok navn: Er CO2 ppm <= 800

For at kontrollere om CO2 i ppm er mindre eller lig 800 bruger vi et elif-statement som kontrolstruktur. Hvis er mindre eller lig 800 går forsætter programmet til "Tænd LED-ring Gul farve" blokken. Vi bruger elif for at sikre at programmet ikke prøver at tænde både gult og grønt lys på samme tid.

Blok navn: Tænd LED-ring Gul farve

For at tænde neopixel led ringen i grøn har vi fulgt samme princip som "Sluk LED-ring" blokken. Her har vi bare lavet en funktion vi kalder np_yellow() som sender rgb værdien (50, 50, 0) til neopixel LED-ringen.

Blok navn: Else

Til sidst bruges et else statement, der aktiverer, hvis både if og elif statements ikke var sande, altså der er mere end 800 CO₂ ppm i luften. Hvis CO₂ i ppm er oversteget 800 køre programmet "Tænd LED-ring rød farve" Blokken.

Blok navn: Tænd LED-ring rød farve

For at tænde neopixel led ringen i grøn har vi fulgt samme princip som "Sluk LED-ring" blokken. Her har vi bare lavet en funktion vi kalder np_red() som sender rgb værdien (50, 0, 0) til neopixel LED-ringens.

Blok navn: Publish ppm til MQTT-broker

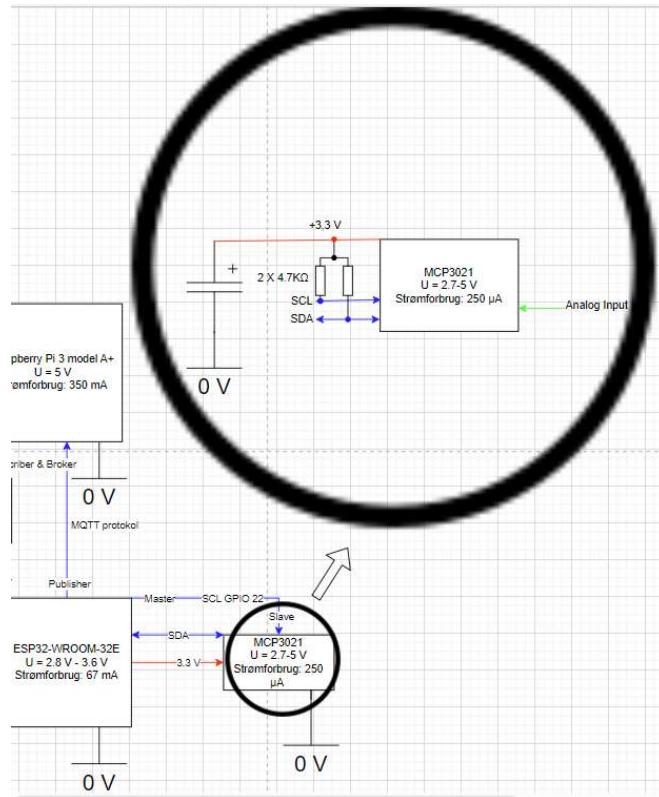
For at sende ppm til MQTT-brokeren bruger vi .publish fra MQTTClient. Vi bruger adc_read() funktionen vi har lavet tidligere til at aflæse målingen, adc_to_ppm() funktionen til at omregne adc målingen til CO₂ i ppm. Herefter bliver CO₂ i ppm formateret til binær og send med .publish().

Blok navn: Sleep 5 minutter

Programmet sættes på pause i 5 minutter.

8. Løsningsdesign

8.1 Elektriske skematikker, målinger og udregninger (Indlejrede systemer)



Figur 3 MCP3021 kredsløb

På figur 3 bliver der fremvist hvordan vi har anvendt vores MCP3021, med modstande og kondensator, samt den serielle forbindelse til vores ESP32.

Målingerne i vores DUT er således:

Strømförbrug:

MQ135: 150 mA

NeoPixel ring: 48 mA

Raspberry Pi 3 Model A+: 350 mA

ESP-WROOM-32E: 67 mA

MCP3021: 250 μA

$$48 \text{ mA} + 150 \text{ mA} + 350 \text{ mA} + 67 \text{ mA} + 250 \mu\text{A} = 615,25 \text{ mA}$$

$$\text{Effektforbrug: } 0,61524 \text{ A} * 5 \text{ V} = 3,07 \text{ W}$$

8.2 Kode beskrivelse (Programmering)

```

7  from time import sleep
8  import network
9
10 def do_connect():
11
12     sta_if = network.WLAN(network.STA_IF)
13     if not sta_if.isconnected():
14         print('connecting to network...')
15         sta_if.active(True)
16         sta_if.connect('MSI7660', 'Kdmmmy5c#')
17         while not sta_if.isconnected():
18             pass
19         print('network config:', sta_if.ifconfig())
20
21 sleep(60)
22 do_connect()

```

Figur 4 kode eksempel 1 – linje 7 til 22 i boot.py på ESP32

I kode eksempel 1 fra boot.py på ESP32 importere vi først sleep() funktionen fra time modulet og modulet network. På linje 10 i koden definerer vi funktionen vi kalder do_connect(). Herefter laver vi en instans a network.WLAN(network.STA_IF) vi kalder sta_if. Denne instans gør, at vi med vores wifi interface kan forbinde til et netværk. På linje 12 kontrollere vi om vores ESP32 har fået forbindelse til netværket. Hvis ikke den har netværksforbindelse bruger vi print() funktionen til at printe besked'en "connecting to network...". På linje 15 aktivere vi vores interface ved at skrive sta_if.active(True), på linjen under bruger vi .connect() metoden og giver den det første argument som er SSID'et på det netværk vi vil oprette forbindelse til og et andet argument som er kodeordet. While-loopet køre funktionen indtil vores station interface har oprettet forbindelse. Når forbindelsen er blevet oprettet, vil der blive printet "network config: vores IP-adresse, subnetmaske, IP-adressen til vores gatewayrouter". På linje 21 bruger vi sleep(60) funktionen til at pause boot i et 60 sekunder. Først på linje 22 køre vi vores do_connect() funktion.

Se hele koden under bilag 5.

```

62 # Funktion til tråd 1
63 def sensor_indikator():
64     while True:
65         adc_value = read_adc()
66         co2_ppm = adc_to_ppm(adc_value)
67         print(f"CO2 Level: {co2_ppm} ppm")
68         sleep(1)

69         if co2_ppm <= 500:
70             np_Green()
71
72         elif co2_ppm <= 800:
73             np_Yellow()
74
75         else:
76             np_Red()
77

```

Figur 5 kode4 eksempel 2 linje 62 til 77 i main.py på ESP32

I kode eksempel 2 fra main.py på ESP32 definere vi en funktion vi kalder sensor_indikator() som senere i programmet vil blive startet på sin egen tråd. Funktionen indeholder et while-loop, der er sat til at være True. Det gør, at funktionen vil køre konstant når først den bliver kaldet. På linje 65 laver vi variablen adc_value som indeholder funktionen read_adc() der aflæser vores adc. På linje 66 laver vi variablen co2_ppm som indeholder funktionen adc_to_ppm(adc_value), som omregner de adc værdier vi får fra adc_value til CO2 i ppm. På linje 67 printer vi en f-string der viser "CO2 Level: {co2_ppm} PPM". Ved at bruge f-string kan vi putte vores co2_ppm variabel ind i og få det printet. Fra linje 70 begynder vores kontrolstrukturer, først med et if-statement, der kontrollerer om co2_ppm er mindre eller lig med 500. Hvis dette er sandt, bliver der på linje 71 kørt funktionen np_Green(), som tænder alle LED'er på neopixel i grøn. På linje 73 bruger elif til at kontrollere om co2_ppm er mindre eller lig med 800. Ved at bruge elif sikre vi at det den indrykkede kode på linje 74 kun køre hvis vores første if-statement ikke er sandt. Hvis co2_ppm altså er mindre eller lig 800 og større end 500 vil np_Yellow() som tænder guld farve på neopixel køre. Hvis ingen af de overstående kontrolstrukture er sande vil else: på linje 76 køre funktionen np_Red() som tænder rødt lys.

Se hele koden under bilag 6.

```

79 | # Funktion til tråd 2
80 | def mqtt_publish_til_RPI():
81 |     while True:
82 |         adc_value = read_adc()
83 |         co2_ppm = adc_to_ppm(adc_value)
84 |         msg = (b'{}`).format(co2_ppm)
85 |         client.publish(TOPIC, msg) # Publish sensor data to MQTT topic
86 |         print(msg)
87 |         sleep(30)

```

Figur 6 5 kode eksempel 3 linje 79 til 87 i main.py på ESP32

I kode eksempel 3 fra main.py på ESP32, definere vi funktionen mqtt_publish_til_RPI() som senere i programmet vil blive startet på sin egen tråd. Funktionen indeholder et while-loop som er sat til at være True. På linje 82 har vi lavet samme variable som på linje 65 i kode eksempel 2. Linje 83 er samme variabel som linje 66 i kode eksempel 2. På linje 84 laver vi en variabel kaldet msg som indeholder formateret string der formaterer co2_ppm til bytes. På linje 84 bruger vi .publish() som er en metode fra MQTTCClient klassen der er blevet importeret fra umqtt.simple modulet. Det første argument vi giver metoden, er TOPIC, det andet argument er variablen msg som vi lavede tidligere i funktionen. Efterfølgende printer vi msg, og bruger sleep(30) funktionen til at pause loopet i 30 sekunder.

Se hele koden under bilag 6.

```

8 | # log sensor data on database
9 | def log_data(ppms):
10|
11|     conn=sqlite3.connect(db_name)
12|     curs=conn.cursor()
13|     curs.execute("INSERT INTO MQ135_data values(datetime('now'), (?))", (ppms,))
14|     conn.commit()
15|     conn.close()

```

Figur 7 6 kode eksempel 4 linje 8 til 15 log_MQ135.py på Raspberry pi 3 A+

I kode eksempel 4 fra log_MQ135.py på Raspberry Pi 3 A+, definere vi funktionen log_data() som skal have et argument. I funktionen bliver der brugt .connect(), .cursor(), .commit() og .close som alle er fra sqlite3 modulet i python. Derfor har vi tidligere i koden importeret sqlite3. På linje 11 laver vi en instans af sqlite3.connect(db_name) som vi kalder conn, som refererer til vores forbindelse til databasen. Argumentet db_name refererer til navnet på den database vi vil forbinde til. På linje 12 laver vi en instans af conn.cursor() som vi kalder curs, den gør det muligt at skrive til databasen. På linje 13 bruger vi curs.execute("INSERT INTO MQ135_data values(datetime('now'), (?))", ppmd,)). Her skriver vi til databasen at den skal indsætte den nuværende dato og tidspunkt samt ppms ind i

dens MQ135_data table. På linje 14 bruger vi conn.commit() som sikre at det vi har sat ind i vores table bliver gemt. På linje 15 bruger vi conn.close() til at lukke forbindelsen til databasen.

Se hele koden under bilag 3.

```

22 | # Callback fires when a published message is received.
23 | def on_message(client, userdata, msg):
24 |     # Decode ppm of co2 values from binary message payload.
25 |     ppm = [float(x) for x in msg.payload.decode("utf-8").split(',')]
26 |     # print('{0} ppm'.format(ppm))
27 |     ppms = ppm[0]
28 |     ppm_int = int(ppms)
29 |     # print(ppm_int)
30 |     log_data(ppm_int)

```

Figur 87 kode eksempel 5 linje 22 til 30 i log_MQ135.py på Raspberry pi 3 A+

I kode eksempel 5 fra log_MQ135.py på Raspberry Pi 3A+, definere vi funktionen on_message(client, userdata, msg). Her bliver det sidste argument msg brugt til at udtag CO2 i ppm. På linje 25 laver vi en variabel, der afkoder den binære payload til utf-8 og putter i en liste. Her er .payload attributten fra paho.mqtt.client modulet blevet kombineret med den indbyggede decode() funktion. På linje 27 laver vi en variabel kaldet ppms som indeholder indholdet af det, der står først i ppm listen. På linje 28 sikre vi os at det vi har i variablen ppms er et integer ved at bruge funktionen int(ppms) og gemme den i variablen ppm_int. På linje 30 bruger vi funktionen log_data() med argumentet ppm_int til at logge dataen til databasen.

Se hele koden under bilag 3.

```

9 | import sqlite3
10 | conn=sqlite3.connect('../sensor_data.db', check_same_thread=False)
11 | curs=conn.cursor()
12 |
13 | # Modtag den nyeste data fra databasen
14 | def get_last_data():
15 |     for row in curs.execute("SELECT * FROM MQ135_data ORDER BY timestamp DESC LIMIT 1"):
16 |         time = row[0]
17 |         ppm = row[1]
18 |         print(time)
19 |     #conn.close()
20 |     return time, ppm

```

Figur 9 8 kode eksempel 6 linje 9 til 20 i MQ135_web.py på Raspberry Pi 3A+

I kode eksempel 6 fra MQ135_web.py på Raspberry Pi 3 A+, har vi ligesom i log_MQ135.py importeret sqlite3 og lavet en instans af .connect() og .cursor(). I MQ135_web.py har vi dog givet .connect det ekstra argument "check_same_thread=False", det gør det muligt for vores python program at forbinde til sqlite3 databasen selvom den ikke er i samme tråd. På linje 14 definere vi

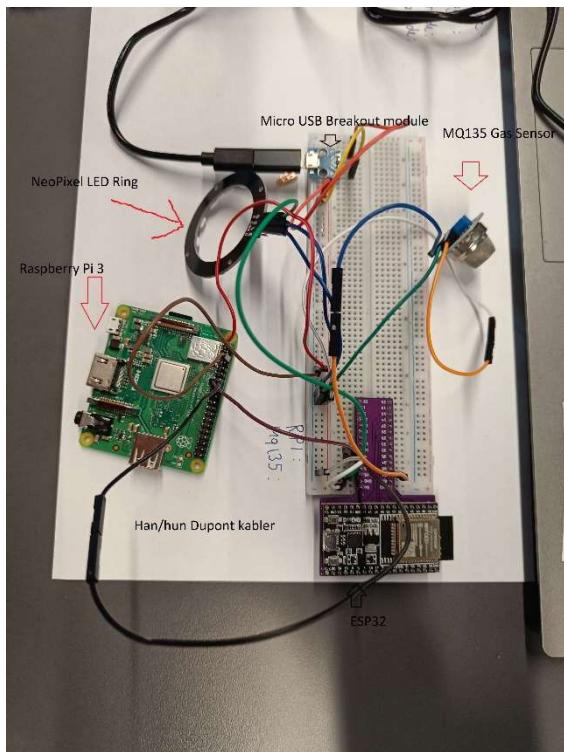
funktionen `get_last_data()` der skal bruges til at hente den nyeste data fra databasen. På linje 15 laver vi et for loop der iterere gennem alle rows på vores `MQ135_data` table i databasen. På samme linje bruger vi så `.execute()` metoden til at eksekvere SQLite kommandoen "`SELECT * FROM MQ135_data ORDER BY timestamp DESC LIMIT 1`" det gør at vi tager alt fra `MQ135_data` sætter det i en rækkefølge hvor den nyeste tid er først og kun tager den første og nyeste data. Herefter laver vi på linje 16 en variabel kaldet `time` som indeholder `timestamp` fra databasen ved at skrive `row[0]`. På linje 17 laver vi variablen `ppm` som indeholder dataen fra `row[1]`. Herefter bruger vi `print()` funktionen til at printe indeholdet af `timevariablen` på linje 18. Til sidst i funktionen returnere vi så `time` og `ppm`.

Se hele koden under bilag 4.

9. Test af løsning

9.1 Pathfinding testresultater

Testopstilling

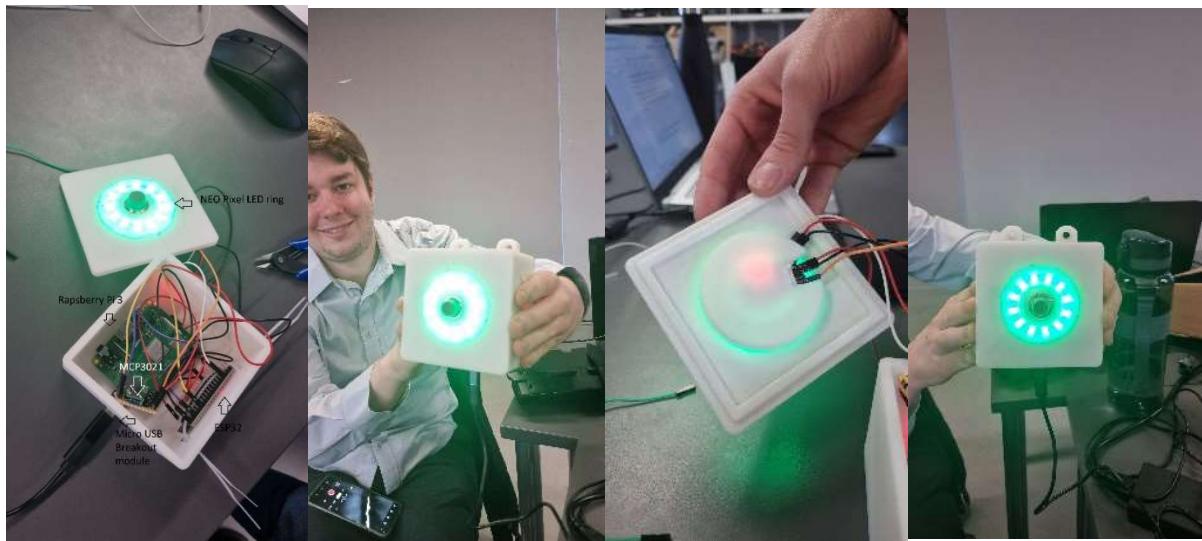


Figur 109 testopstilling

Figur nr. 10 viser hvordan vores testopstilling så ud, da vi skulle teste om vores komponenter og sensorer kunne arbejde sammen. Vi valgte derfor først at samle vores testopstilling på et fumlebræt, i og med at det var simpelt at sætte sammen og tage fra hinanden igen. Derudover skulle vi se om vores usb breakout module kunne forsyne alle de komponenter vi brugte i vores projekt. Her fandt vi ud af at vi godt kunne forsyne hvad vi skulle anvende, og derfra kunne vi gå videre til at lodde vores løsning sammen.

9.2 Dokumentation af DUT

Billeder og video af samlet løsning



Figur 1110 billede af vores DUT

Specifikationer

Stadie:	Nuværende DUT stadie	Ønsket endelig stadie
Højde	6,5cm	4 cm
Bredde,	10x11,5cm	10x10
Længde	163,5cm	57,5 cm
Vægt	302 g	253 g
Indgang spænding	5 V	5V
Effektforbrug	3,07	2,5 W
Netværksprotokoller	MQTT, HTTP	MQTT, HTTP

9.3 Udførsel af Accepttest på DUT

Test af Krav 1:

Kategori: HW	<p>Accept test:</p> <p>Step 1) Løsningen tændes.</p> <p>Step 2) Løsningen rystes i 10 sekunder.</p> <p>Hvis løsningen ikke har taget skade og man ikke kan høre noget inde fra kassen, vurderes kravet som værende bestået.</p>	Det vurderes at kravet er fuldført.
	<p>Diskussion og evt. Ønskede ændring af krav:</p> <p>Accept test 1: Ryste test</p> <p>https://youtube.com/shorts/LYleill5jTY</p> <p>Løsningen blev rystet på forskelligt vis i over 10 sekunder. Ingen raslen kunne høres og intet af løsningen havde taget skade efter test.</p>	

Test af krav 2:

ID: 2	Krav: Løsningen skal kunne give en visuel markør, når CO2 niveauet overstiger den anbefalet mængde i et rum.	Prioritet: 1
Kategori: Visuelt	<p>Accept test:</p> <p>Step 1) Løsningen tændes.</p> <p>Step 2) Der åndes på sensoren indtil den visuelle indikator skifter farve til rød.</p> <p>Hvis måleren kommer op over 800 PPM og at den visuelle markør bliver skiftet til farven rød, vurderes kravet som værende bestået.</p>	Det vurderes at kravet er fuldført.
	<p>Diskussion og evt. Ønskede ændring af krav:</p> <p>Accept test 2: Ånde test</p> <p>https://youtube.com/shorts/IyUEk3eyWxY</p> <p>Efter at have åndet på sensoren i omkring 16 sekunder skiftede måleren til rød.</p> <p>Det kan være vanskeligt at puste CO2 på sensoren. Eventuelt kan testen ændres til, at løsningen bliver placeret i en lukket kasse, og så puster man i den til den lyser den ønskede farve.</p>	

Test af krav 3:

ID: 3	Krav: Løsningen skal kunne forbindes til et lokalnetværk.	Prioritet: 1
Kategori: Forbindelse	<p>Accept test:</p> <p>Step 1) Løsningen tilsættes strøm forsyning.</p> <p>Step 2) Efter 1 minut skal løsningen være tilkoblet lokalnetværket.</p> <p>Step 3) Løsningen pinges fra en computer.</p> <p>Hvis løsningen har forbindelse til lokalnetværket efter 1 minut og ingen ping-pakker går tabt, vurderes kravet som bestået.</p>	Det vurderes at kravet er fuldført.
	<p>Diskussion og evt. Ønskede ændring af krav:</p> <p>Accept test 3: Ping test</p> <p>https://youtube.com/shorts/xozBPMrtbAo</p> <p>Efter at have pinget følgende IP: 192.168.137.65, fik vi svar med det samme fra den på alle 4 "test" pings.</p>	

Test af krav 4:

ID: 4	Krav: Kunne vise en graf over CO2 niveau på en hjemmeside.	Prioritet: 1
Kategori: Programmering	<p>Accept test:</p> <p>Step 1) Løsningen tændes.</p> <p>Step 2) En computer opretter forbindelse til løsningens hjemmeside.</p> <p>Step 3) Der kontrolleres om hjemmesiden viser en graf der indeholder data fra CO2 måleren.</p> <p>Tallene fra Shell sammenlignes med tallene fra hjemmesiden og tjekker om den viser en passende graf over CO2 PPM, hvis tallene passer, vurderes kravet som bestået.</p>	Det vurderes at kravet er fuldført.
	<p>Diskussion og evt. Ønskede ændring af krav:</p> <p>Efter at have skrevet kommandoen ind for at få tallene frem og sammenlignet dem med hjemmesiden, fik vi præcis samme tal</p> <p>Accept test 4: Graf og SSH-data</p> <p>https://youtube.com/shorts/LQK5187ZBBc</p>	

Test af krav 5:

ID: 5	Krav: Grafen på hjemmesiden skal kunne opdateres hvert 5 minut med ny data fra de sidste 5 minutter	Prioritet: 1
Kategori: Prog	<p>Accept test:</p> <p>Step 1) Løsningen tændes.</p> <p>Step 2) Grafen opdateres ved et tryk på refresh</p> <p>Step 3) Grafen opdateres og tjekkes hvert 5 minutter for opdateringer over CO2 niveauet i lokalet.</p> <p>Hvis der er ny data hvert. 5 minut vurderes kravet som bestået.</p>	Det vurderes at kravet er fuldført.
	<p>Diskussion og evt. Ønskede ændring af krav:</p> <p>Accept test 5: Opdatering af graf</p> <p>https://www.youtube.com/shorts/q4PYzU3lsPl</p> <p>Testen blev fuldført og blev opdateret manuelt efter 23 minutter.</p>	

Test af krav 6:

ID: 6	Krav: Løsningen skal have dokumentation på at den er testet i minimum 3 timer	Prioritet: 1
Kategori: Power	<p>Accept test:</p> <p>Step 1) Løsningen bliver sat op, og tilsluttet strømkilde.</p> <p>Step 2) Timer startes på 3 timer</p> <p>Når ingen problemer opstår efter 3 times brug, vurderes kravet som bestået.</p>	Det vurderes at kravet er fuldført.
	<p>Diskussion og evt. Ønskede ændring af krav:</p> <p>Accept test 6: 3 timers test</p> <p>https://www.youtube.com/watch?v=IkizqBR4JLY&t=2137s</p> <p>Løsningen forblev tændt under alle 3 timer og virkede som forventet.</p> <p>Vi kan også konstatere at luft kvaliteten i lokalet er rimelig god, da LED'en primært lyste grønt igennem hele testen.</p>	

Test af krav 7:

ID: 7	Krav: Løsningen skal kunne forbinde trådløst til en mikrocontroller.	Prioritet: 2
Kategori: Forbindelse	<p>Accept test:</p> <p>Step 1) Løsningen & mikrocontroller tændes</p> <p>Step 2) Løsningen sender besked til mikrocontroller trådløst</p> <p>Hvis besked bliver verificeret som det der blev afsendt, vurderes kravet som bestået.</p>	Det vurderes at kravet ikke er fuldført.
	<p>Diskussion og evt. Ønskede ændring af krav:</p> <p>I og med at vi ikke fik klaret dette krav, er næste skridt i processen at få det lavet. Vi havde desværre problemer tidligere i projektet som tog den tid vi havde planlagt til 2. prioriteterne.</p>	

Test af krav 8:

ID: 8	Krav: Løsningen skal være i stand til at åbne et vindue, ved for højt CO2 niveau i rummet.	Prioritet: 2
Kategori: HW/Prog	<p>Accept test:</p> <p>Step 1) Løsningen tændes</p> <p>Step 2) Der åndes på måleren til PPM når >800</p> <p>Hvis måleren sender signal og åbner vinduet ved hjælp af motoren, vurderes kravet som bestået.</p>	Det vurderes at kravet ikke er fuldført.
	<p>Diskussion og evt. Ønskede ændring af krav:</p> <p>ID 8 kravet er ligesom det tidligere krav, 7, som vi desværre ikke fik nået at lave inden deadline.</p>	

10. Praktisk projektplanlægning og ledelse

10.1 WBS

WBS (Work Breakdown Structure) er en projektledelses metode man bruger til at skabe et overblik over projektet. Man ser på alt man skal igennem for, at projektet bliver en succes. Projektet bliver brutt ned til mindre opgaver, som man skal igennem for at nå målet. I vores projekt var det den første projektledelses model vi gjorde brug af. Vi startede med at snakke om de større opgaver vi skulle igennem og herefter fandt vi underopgaverne til hver større opgave. I projektet var WBS en god hjælp til at få et hurtigt og godt overblik over hvad projektet egentlig indebar. I løbet af vores projekt måtte vi ændre en smule i vores WBS, da der både kom flere ting, der skulle løses og nogle ting, der skulle fjernes.

10.2 Gantt

Vi har kørt en anderledes version, der tager sprint'ne fra Scrum og sætter dem ind i Gantt. Dette har givet os mulighed for at gøre det nemmere visuelt til hver Scrum møde og at den har været nyttig til at give et bedre visuelt overblik over deadlines, sprints & hvem der arbejder på hvad. Dette gjorde at vores Scrum møder blev mere strukturerede og effektive.



Figur 12 Gantt diagram for 1. sprint

På figur 12 ses vores første sprint der var indsat i Gantt diagrammet. I først række (Fra venstre og ned) ses 1. Sprint titlen som indikerer hvilket sprint vi er i og derefter de første opgaver der var blevet planlagt for det første sprint. Her bliver der blandt andet inkluderet de startende opgaver som at få lavet indledende research samt at opsætning af backlog.

I anden kolonne ses vurderingen af risikoen af opgaven. Risikoen bliver vurderet ud fra sværhedsgrad af opgaven, deadline på opgaven og størrelse på opgaven. Efter at have fået sin risiko ”titel” tildelt får den automatisk en farve (Lys blå = lav risiko / Mørk blå = mellem risiko / Lilla = høj risiko). Ved første sprint var vi forholdsvis sikre på altting ville blive klaret forholdsvis let grundet de indledende opgaver er mindre tidskrævende og er fundamentet for at vi kan fortsætte med projektet.

I tredje kolonne ses navnene på dem opgaven er blevet tildelt til, derudover er der også ”alle” hvilket betyder det er en opgave vi lave fælles som gruppe.

I fjerde kolonne ses et % tal der udgør hvor meget af opgaven man er færdig med. Dette bliver også brugt agilt og bliver opdateret undervejs.

I femte kolonne kan start dato ses. Dette er den allokeret dag hvori opgaven bliver i gang sat.

I sjette kolonne bliver der sat x mængde dage af til hver opgave. Dette bliver planlagt ud fra hvor meget tid personen med ansvaret for opgaven føler de har brug for. Dette planlægges og aftales normalt ved det daglige morgen Scrum møde.

Igennem projektet har Gantt og Scrum været en vital del af gruppens projektledelse. Gantt diagrammet er blevet brugt og opdateret under hver Scrum møde der er blevet holdt. Under projektet skete der afvigelser fra den originale tidsplan hvilket vi effektivt kunne håndtere på grund af vores projektledelse stil.

3. Sprint					
Kravspec	Mellem risiko	Emil, Oliver & Christian	80%	16-03-2023	1
HW-Blok Diagram	Høj risiko	Oliver	50%	17-03-2023	2
Flowchart	Mellem risiko	Magnus	100%	17-03-2023	1
SW-Blok Diagram	Høj risiko	Magnus & Christian	0%	18-03-2023	1
BOM - Liste	Mellem risiko	Daniel	100%	16-03-2023	3



Figur 13 Gantt diagram fra 3. sprint

I vores tredje sprint løb vi ind i et tidsmæssigt problem og vi fejl estimerede hvor lang tid den planlagte opgave ville tage eftersom vi fik nogen rettelser fra vores vejledere. Dette betød for os som gruppe at vi var nødt til at bringe HW-Blok Diagram og SW-Blok Diagram tilbage til backlog og have den med til næste sprint.

10.3 Scrum

Scrum er en Agil projektledelsesmetode, hvor der er 3 forskellige roller. Den første rolle er development team, som er det hold der arbejder på projektet. Herefter er der Scrum master, som er lederen af development teamet. Scrum masteren er den person, der har kontakt til Product Owner. Product Owner er den person, der ”ejer” projektet, det er ham scrum master og development teamet referer til. Scrum består af 6 forskellige dele. Den første del er Produkt-Backlog, som er den del, hvor man lister alle de ting, som projektgruppen skal igennem for at komme i mål med projektet. Den næste del man møder i Scrum er Sprint-backlog. Sprint-backlog er det man skal nå i løbet af det sprint man er i gang med, vores sprint havde en varighed af 3 dage pr. sprint. Det er her Scrum er anderledes end KANBAN tavlen, hvor der ikke er sprints i løbet af projektet.

Efter Sprint-backlog møder man ”TO DO” det er alle de opgaver, der ikke er påbegyndt, som skal laves. Del 4 er ”doing”, det er alle de opgaver fra vores produkt-backlog, som, der er i gang i det nuværende sprint. Test er de ting, som muligvis er færdige eller mangler godkendelse fra en Product Owner. Den sidste del af Scrum er ”Done” som er alle de ting der er færdige og/eller godkendt af projektets’ product owner. I Scrum er der inkorporeret møder før efter og under et sprint.

I Sprint planlægningsmødet, planlægger man, hvad der skal laves under det kommende sprint. Under et sprint holder, man et dagligt scrum møde, hvor man opdaterer hinanden og finder ud af om sprintet går efter planen på den på gældende dag.

Efter et sprint er ovre, har man et sprint retrospektive møde, det er her man finder ud af om alt det man skulle nå i løbet af sprintet, er blevet lavet og om nogle af tingene trak ud og skal med på næste sprint igen.

Scrum var en rigtig god måde at holde styr på projektet. Vi kunne holde overblikket over de forskellige opgaver og de daglige møder var en stor hjælp til at holde styr på hvor langt vi egentlig var i processen. Det retrospektive møde var vigtigt for os, da vi hurtigt fandt ud af, hvilket ting der fungerede og ikke fungerede i de forskellige sprints vi kom igennem.

10.4 Projektanalyse

10.4.1 Risikoanalyse

I projektet brugte vi en Risikoanalyse til at komme problemer og udfordringer i forkøbet. I starten af projektet blev vi enige om hvilke udfordringer vi kunne komme til at møde. Herefter gav vi dem point fra 1-5 ud fra hvor stor en konsekvens det kunne have for projektet. Sandsynligheden for at vi mødte de valgte udfordringer blev også vurderet fra 1-5. Efter både konsekvens og sandsynlighed fik en vurdering mellem 1-5 lagde man de 2 tal sammen og gav dem en samlet rating. Efter man har fået et samlet antal point, kommer man med en forebyggende foranstaltning for at have en løsning klar på forhånd, så man ikke først skal til at finde ud af hvad man gør, når problemet opstår. Ved hvert punkt i risikoanalysen er der altid en der har ansvaret for, at den forebyggende foranstaltning bliver opretholdt i tilfælde af, at det bliver nødvendigt.

Hvad kan der ske?	Konsekvens 1-5 (K)	Sandsynlighed 1-5 (S)	Risikopoint (K) x (S)	Forebyggende eller afhjælpende foranstaltning	Ansvarlig

Forkert/fejl opstilling af elektriskkredsløb der destruerer vitale komponenter	4	3	12	<p>Forebyggende:</p> <p>Man skal sikre sig at komponenter ikke bliver forsynet med en større spænding end de kan tåle. Derudover skal man sikre at indgangsben på ESP32 eller Raspberry PI ikke bliver forsynet med for høj spænding på deres signal ben. Dette gøres ved at konsultere Datablad for komponenter, læse HW-Blokdiagram og kredsløbstegning.</p> <p>Afhjælpning:</p> <p>Bestilling af nye komponenter så frem KEA ikke har dem til rådighed skal ske lige så snart man ved at en komponent ikke er funktionsdygtig.</p>	Alle
Sygdom i projektgruppen	3	3	9	Forebyggende:	Alle

				<p>Projektgruppen sørger for at opretholde normal hygiejne</p> <p>Afhjælpning:</p> <p>Hvis projektgruppemedlemmets opgave i sprintet, er af en højere prioritet end resten af projektgruppens opgave. Overtages medlemmets opgave af en anden i projektgruppen.</p>	
Forsinkelser i levering af løsningens komponenter	3	3	9	<p>Forebyggende:</p> <p>Hvis komponenter skal bestilles, gøres dette så tidligt i projektet som muligt.</p> <p>Afhjælpning:</p> <p>Hvis en komponent er så forsinket, at det vil sænke projektgruppens fremgang. Skal denne</p>	Alle

				komponent, så frem det er muligt, bestilles fra en anden leverandør.	
Problemer/splid og uenigheder i projektgruppen	2	3	6	<p>Forebyggende:</p> <p>Det er projektgruppemedlemmernes ansvar at tage bekymringer op til de daglige SCRUM-møder. På denne måde kan vi imødekomme hindanen og skabe et godt arbejdsmiljø.</p> <p>Afhjælpning:</p> <p>Skulle der forekomme problemer, splid eller uenigheder i projektgruppen, skal dette snakkes igennem. Hvis ikke der kan skabes et kompromis må lære/vejledere spørges til råds.</p>	Alle
	5	1	5	Forebyggende:	Alle

IT-udstyr går i stykker				<p>Det er projektgruppemedlemmernes eget ansvar at have funktionsdygtigt IT-udstyr.</p> <p>Afhjælpe:</p> <p>Man skal så hurtigt som muligt have repareret eller anskaffet sig nyt IT-udstyr.</p>	
Miste indtil videre færdiglavet løsning (software/hardware)	5	1	5	<p>Forebyggende:</p> <p>Projektgruppen skal sikre at der bliver lavet versioneringer, og backup af løsningens software del. Løsningens hardware del skal opbevares i projektgruppens tildelte skab aflåst.</p>	Alle

10.4.2 Interessentanalyse – IoT 2:

Gidsel: Skoleelever	Resurse person: Projektgruppen.
-------------------------------	---

Skolelærerne	Forskere. Rektoren af skolen. Lærere som bliver involveret i spørgeskema.
Ekstern interessent: Skoleelevernes forældre. Andre Folkeskoler uden for projektet.	Grå eminence: Kommunen (Københavns Kommune), der vil være dem der implementerer idéen i folkeskoler.

Vi har valgt skoleelever som gidsler, fordi de ikke har en indflydelse på om idéen bliver implementeret, og de har ikke noget at sige om hvorvidt det skal laves eller ej, men det er dem der bliver påvirket mest af, at idéen bliver gennemført i deres klasselokaler.

Resursepersonerne i dette projekt er; Projektgruppen som står for projektet, idéen og løsningen. Forskere der har indledt forskning og test der har hjulpet med at finde informationer og statistikker der viser hvor skadeligt et dårligt indeklima er, samt hvad der kan gøres for at forhindre dette. Rektoren af skolen er en resurse person på i og med at det er ham der skal implementere det på skolen, på baggrund af en forespørgsel fra den Grå Eminence (Københavns Kommune). I dette projekt er skolelærere gidsler, fordi de ikke har indflydelse på implementeringen af idéen, men de bliver alligevel en resurse person, på baggrund af deres deltagelse i det spørgeskema vi har sendt ud til dem.

De eksterne interesserter i projektet er skoleelevernes forældre, på baggrund af at de får informationen om implementeringen af idéen, fra deres børn (gidslerne i projektet). Derudover har vi valgt andre folkeskoler end den vi har udvalgt i projektet, som vil have mulighed for senere hen at få implementeret idéen, men på nuværende tidspunkt ikke har noget med projektet at gøre.

Den grå eminence er Københavns Kommune, grundet at det er dem der går efter planen om at få mindre CO2 udslip i skolerne, samt at få et bedre indeklima på deres skoler. Det vil være dem, der er Product Owner i projektet.

11. Konklusion

Dårligt indeklima er et problem i folkeskolen og har været det igennem mange år. Kommuner har lavet mange undersøgelser igennem årene, men specielt højt CO₂ i luften er et problem, der fortsat plager skolerne. Vores mål var at lave en IOT-løsning, der kunne forbedre indeklimaet og det har vi gjort ved at lave en løsning, der mäter CO₂ indholdet i luften og viser det visuelt via en LED ring og en hjemmeside. Vi fik dog ikke færdig gjort den endelige løsning, der automatisk skulle åbne et vindue.

Den største grund til dårligt indeklima i folkeskolen er dårlig ventilation, det kan være i form af, at åbne vinduer og døre eller at have et dyrt ventilationsanlæg.

Dårligt indeklima har mange konsekvenser og det bliver værre desto dårligere indeklimaet er. Sløvhed og træthed er det hyppigste problem, som skoler dører med på grund af høj CO₂ i luften. Hovedpine og virusser er også problemer som dårligt indeklima er med til at forårsage.

MQ135 Gas sensor kan mæle CO₂'en i luften og da et højt CO₂ indhold i luften typisk betyder dårligt indeklima, så er det en god og billig metode til at mæle forholdene, der skaber et dårligt indeklima.

Sensoren indsamler, hvor meget CO₂, der er i luften i formatet PPM og det kan indsættes i en database, så man kan se hvor god luftkvaliteten er i løbet af dagen. Vi har også lavet en grafisk fremstilling i form af en graf på en hjemmeside.

Vi kan konkludere, at vi har lavet en IOT-løsning, der kan mæle dårligt indeklima, men har ikke udfyldt vores anden prioritet, der automatisk ville være med til at løse problemet. Løsningen vil stadig hjælpe med problemet, men det vil være op til lærere, elever og skoler at gøre noget ved problemet som løsningen viser via visuelle indikator som vores LED ring og hjemmeside.

12. Projektforløbet

I projektet var samarbejdet godt og fokuseret. Vi havde stor fokus på projektledelsen, for at få et godt overblik over hvad der blev arbejdet på. Dette gjorde vi ved at anvende forskellige projektledelses værktøjer, såsom WBS, GANTT, og SCRUM. Dette gjorde at de forskellige opgaver gruppemedlemmerne fik, var transparente og alle vidste hvad der på et givent tidspunkt blev arbejdet på. Derudover var gruppen gode til at spørge om hjælp, hvis der var problemer eller forhindringer i ens arbejde, hvilket hjalp på effektiviteten af arbejdet og en stor besparelse på tid.

Der var dog nogle få kommunikationsproblemer i projektet. Dette var dog ikke så stort et problem, i og med at konsekvenserne kun var, at man arbejdede forskellige steder. Problemet blev løst ved at vi begyndte at holde SCRUM-møder når vi var færdige med at arbejde. Vi havde altså et møde om morgenen, så vi vidste hvad der skulle laves hen af dagen og hvilke opgaver der var uddelt. Mødet senere blev brugt til at finde ud af hvad vi nåede den dag, og hvad vi kunne forvente at lave dagen efter, samt hvornår og hvor vi skulle mødes.

I fremtidige projekter skal vi være bedre til at sørge for en god kommunikation fra starten af projektet, så der ikke sker misforståelser. Bedre kommunikation fra starten af kunne også have resulteret i, at vi havde fået vores 2. prioriteter. Derudover var der også misforståelser med hvordan en del af projektet skulle laves, og det resulterede i at der blev brugt en hel dag på noget vi ikke kunne anvende alligevel. Dette kunne have været forhindret, hvis vi brugte længere tid på at læse og forstå projektbeskrivelsen.

13. Perspektivering

I dette projekt er vores 2. prioriteter ikke blevet opnået, så det ville være oplagt at det skulle laves som næste skridt i løsningen. Her vil automatiseringen af vores løsning blive lavet.

Udover at måle CO₂, er der flere parametre man kan måle når det kommer til indeklimaet. Her vil det være muligt at kigge på andre faktorer der altså hæmmer indeklimaet i klasselokalerne. Dette kunne for eksempel være temperaturen i lokalet, der har en relativ stor effekt på hvordan eleverne har det. Selv et lille skifte i temperatur kan give hovedpine og utilpashed.¹¹ Derudover vil man kunne måle på andre forhold, såsom støjniveau der også kan være til gene for nogle elever under undervisningen.

I fremtiden vil det også være muligt at få succes med vores 2. prioriteters kravspecifikationer, der vil automatisere udluftningsprocessen ift. CO₂ ppm. Dette, vil i forhold til vores spørgeskema, hjælpe folkeskolelærerne og altså være med til at få et bedre indeklima i klasselokalerne.¹² Løsningen vil altså ultimativt hjælpe med koncentrationen i undervisningen, og dermed hjælpe til at øge klassens faglige niveau.

Udover at automatisere åbning af vinduer i klasselokaler, ville det være oplagt at automatisere ventilation i lokalet. Dette vil gøre vores løsning mere brugervenlig året rundt uanset årstid, da det kan skabe visse komplikationer for brugerne ved åbning af vinduer om vinteren og ved dårlige vejrfordhold.

Udover at nå vores 2. prioriteter, ville vi også have brugt en del tid på at gøre designet af selve løsningen mere tiltrækkende. Kassen ville blive lavet fladere, da der ikke er behov for så meget plads og dermed blive mere robust og brugervenlig. Ledningerne i boksen skulle være mere organiseret, så indholdet bliver mindre uoverskueligt. Man kunne gøre den mere kosteffektivt, hvis man kunne finde en anden billigere enhed end Raspberry Pi 3 A+ til at køre løsningen på.

¹¹https://www.exhausto.dk/~media/Global/PDF/Institute/Research/Officiel%20Rapport%20resultatrapport_mx2014_5_small.pdf

¹² Se spørgeskemaet i bilag 8.

14. Litteraturliste

Internet artikler:

Andersen, Birgitte; Clausen, Geo og Ørsted-Jordy, Laura: "*MasseEksperiment 2014 Indeklima i Klasselokaler - resultater*",

https://www.exhausto.dk/~media/Global/PDF/Institute/Research/Officiel%20Rapport%20resultatrapport_mx2014_5_small.pdf (Besøgt: 2023-04-02)

Carr, James: "*Classroom carbon dioxide levels up to three times above watchdog guidelines*",

<https://schoolsweek.co.uk/classroom-carbon-dioxide-levels-three-times-above-watchdog-guidelines/> (2023-04-02)

Lauritsen, Helle: "*Masseeksperimentet: CO2-niveauet i klassen sprænger skalaen*",

<https://www.folkeskolen.dk/arbejdsliv-borneliv-fureso-kommune/masseeksperimentet-co2-niveauet-i-klassen-spraenger-skalaen/1377365> (Besøgt: 2023-04-02)

Grynberg, Stine: "*Du bliver dummere i dårlig luft*", <https://www.folkeskolen.dk/arbejdsliv-folkeskolen-nr-09-2019-fysisk-arbejdsmiljo/du-bliver-dummere-i-darlig-luft/1375839> (Besøgt: 2023-04-02)

"*Creating Healthy Indoor Air Quality in Classrooms*",

<https://www.co2meter.com/blogs/news/indoor-air-quality-monitoring-schools-classrooms> (Besøgt: 2023-04-02)

"*Københavns Kommune vil forbedre indeklimaet i skoler og daginstitutioner*",

<https://cphsolutionslab.dk/news/forbedring-af-indeklimaet-i-skoler-og-daginstitutioner> (Besøgt: 2023-04-02)

"SIKRE ALLE LIGE ADGANG TIL KVALITETSUDDANNELSE OG FREMME ALLES MULIGHEDER FOR

LIVSLANG LÆRING", <https://www.verdensmaalene.dk/maal/4> (Besøgt: 2023-04-02)

15. Bilag

Bilag 1: Persona og userstories

Navn: Frederikke Balder.



<https://www.colourbox.dk/billeder/dame-laereanstalt-roedtop-billeder-46309575>

- **Alder:**
 - 28
- **Job:**
 - Folkeskole lærerinde
- **Påklædning:**
 - Løst siddende skjorter og sweatre.
- **Personlige og professionelle ambitioner:**
 - Hun arbejder for at gøre undervisningen spændende og fange hendes elevers interesse.
- **Problemer og udfordringer:**
 - Kan godt blive nervøs og usikker når hun er i det sociale.
- **Interesser/Hobbyer:**
 - Hun har en urtehave som hun bruger meget tid på.
- **Adfærd på sociale medier:**
 - Tagger sine veninder i Instagram posts, derudover poster hun selv når hun er til fester
- **IT-kompetencer:**
 - PowerPoint.

User story:

Som en folkeskole lærerinde tilbringer jeg mycket tid inde i et klasselokale med få pauser til at gå udenfor. Jo længere hen ad dagen vi kommer, desto mere falder effektiviteten af undervisningen og læringen for både mig og mine elever, blandt andet på grund af den dårlige ventilation og indeklima i klasselokalet.	Story nr. 2
Jeg ønsker en løsning der kan hjælpe med at holde et godt indeklima i klasselokalet både for mine elever og min egen skyld.	Prioritet: 4
Jeg håber, at et bedre indeklima gør at mine elever trives bedre.	Effort: 3

Navn: Björn Frigg-Jarnis Fenrisson



<https://www.colourbox.dk/billeder/dreng-klassevaerelset-uddannelse-billeder-54874472>

- **Alder:**
 - 14
- **Job:**
 - Folkeskoleelelev.
- **Påklædning:**
 - Tøj hans mor køber til ham.
- **Personlige og professionelle ambitioner:**
 - Han vil gerne blive bedre til matematik. Han drømmer om at blive marine biolog.
- **Problemer og udfordringer:**

- Han har svært ved at holde koncentrationen i nogle fag.
- **Interesser/Hobbyer:**
 - Fiske, Marine liv & Stjerner.
- **Adfærd på sociale medier:**
 - Følger med i diverse hav-relateret instagram sider og kan godt lide
- **IT-kompetencer:**
 - Har kun en iPad

User story:

Som elev i folkeskolen har Björn svært ved at koncentrere sig i undervisningen.	Story nr. 3
Han drømmer om at blive marine biolog, men har især svært ved at kunne koncentrere sig i matematiktimerne sidst på dagen. Han søger derfor en løsning der kan hjælpe ham med at forblive koncentreret.	Prioritet: 4
Således at jeg kan koncentrere mig i timen er det nødvendigt for mig at små ting der kan øge koncentrationen, bliver bedre. Heraf er indeklima et simpelt problem der kan blive løst.	Effort: 2

Navn: Adam Severinsen



<https://www.colourbox.dk/billeder/laerer-forelaesning-tutor-billeder-57053117>

- **Alder:**

- 40

- **Job:**

- Musiklærer i folkeskolen

- **Påklædning:**

- Går mest i veste i jordnære farver og skjorter.

- **Personlige og professionelle ambitioner:**

- Ønsker at videregive sin kærlighed for musik.

- **Problemer og udfordringer:**

- Jeg kan have svært ved holde fast i eleverne.

- **Interesser/Hobbyer:**

- Adam samler på vinyler.

- **Adfærd på sociale medier:**

- Ikke nogen eksistens på de sociale medier, udover AULA

- **IT-kompetencer:**

- Kan handle de mest basiske i Office pakken, samt lave musik i musikprogrammer.

User story:

Som musik lærer i folkeskolen forsøger jeg altid at finde nye løsninger til hvordan jeg kan opretholde klassens opmærksomhed.	Story nr. 5
Jeg ønsker en løsning der kan mindske tabet af klassens koncentration. Jeg er både villig til at ændre min undervisnings tilgang, men også villig til at afprøve produkter og it-løsninger der kan hjælpe med dette.	Prioritet: 5
Jeg vil gerne fixe indeklimaet fordi det kan blive et problem at have så mange elever i et enkelt rum, hvilket hurtigt tager folk ud af fokus	Effort: 4

Navn: Brunhildur Fenrisdöttir:



<https://www.colourbox.dk/billeder/pige-teen-teenage-billeder-54378680>

- **Alder:**
 - 11
- **Job:**
 - Studerende/Elev
- **Påklædning:**
 - Fleece og Sweater
- **Problemer og udfordringer:**
 - Svært ved at koncentrere sig i skolen.
- **Interesser/Hobbyer:**
 - Klaver og Musik
- **Adfærd på sociale medier:**
 - Har ikke sociale medier.
- **IT-kompetencer:**
 - Computer og Smartphone

User story:

Som en elev i en folkeskole bruger jeg mange timer i klassen til undervisning, ofte bliver luften tung i løbet af dagen, da vi er mange elever i et mindre klasselokale. Luften i klassen gør det svært at koncentrere sig og være produktiv, da udluftningen er dårlig.

Story nr. 6

Jeg ønsker en løsning til at hjælpe med udluftning og bedre indeklima.	Prioritet: 3
Jeg vil gerne blive til noget men et dårligt indeklima kan ødelægge mit fokus og jeg går hjem med en følelse af ikke at have forstået noget.	Effort: 3

Bilag 2: Business case

Business case – Delmål 4.1.1 – IoT 2

Kort resumé

Vi vil løse delmål 4.1.1 fra de 17 verdensmål ved brug af en sensor der måler CO₂ niveauet i et lokale og har en visuel indikator der fortæller hvornår der skal luftes ud. Derudover kan en af vores forretningsmuligheder automatisere udluftning ved at åbne et vindue når CO₂ niveauet er for højt. Vi forventer at kunne øge elevers evne til at klare koncentrationskrævende opgaver med op til 15%. Samt sælge vores løsning til 70% af folkeskoler. Dog kan der være negativt udbytte i form af stigende varmeregning, sygdom grundet træk og om vores løsning i alle tilfælde vil være nok i sig selv. Projektet vil have en tidshorisont på lidt over en måned. Væsentlige risici kan være om lære er indstillet på at skulle bruge vores løsning og om løsning vil være for besværlig uden en automatisk vinduesåbning.

Begrundelser

Vores projekt og løsning har til formål løse delmål 4.1.1 fra verdensmål 4.¹ Dette vil vi gøre ved at hjælpe grundskoler med at opretholde et godt inde klima. Vi har fundet ud af at blandt andet CO₂ til oxygenforholdet i et rum har indflydelse på hvor produktive vi er som mennesker. For høj CO₂ koncentration i et lokale kan give hovedpine. Vi har også fundet ud af en undersøgelse viser op til 60% øget produktivitet hos kontorarbejder i et lokale med lav CO₂ koncentration.²

Forretningsmuligheder

Gøre noget

Projektet vil have en sensor der måler hvor højt CO₂ niveauet er i lokalet, og have en visuel markør der indikerer om det er over det anbefalede niveau. Når sensoren mäter at niveauet er for højt, vil sensoren snakke med en Mikrocontroller, der er tilkoblet til en aktuator der automatisk vil åbne et vindue, så niveauet kommer ned på det anbefalede niveau. Derudover vil vi have en Mikrocontroller der sender dataene sensoren opsamler, til en database der vil vise en grafisk fremstilling på databasen.

Gøre et minimum

Vores sensor vil have mulighed for at indikere når CO₂ niveauet er for højt, via en LED. Der vil være 1 sensorer i spil, der vil stå så centralt i lokalet som muligt. Her vil der være en mikrocontroller der

sørger for at den visuelle markør vil lyse, når CO₂ niveauet er over det anbefalede niveau. Derudover vil vi have en Mikrocontroller der sender dataene sensoren opsamler, til en database der vil vise en grafisk fremstilling på databasen.

Gøre ingenting

At gøre ingenting ved dette problem, vil ende ud i at koncentrationen i undervisningen vil falde, og kvaliteten af undervisningen vil altså på grund af dette også falde.

Forventet udbytte

Vi forventer at kunne forøge elevernes evne til at løse opgaver der er koncentrationskrævende med op til 15%. Samtidig kan vi se at der i en undersøgelse af 250 klasseværelser i 2017 var en for høj mængde CO₂ i 91% af klasseværelserne.³ Vi forventer at sælge vores løsning til 70 % af folkeskolerne.

Forventet negativt udbytte

Varmeregningen vil kunne komme til at stige i skolerne. Elever placeret tættest på vinduer kan blive udsat for træk og som konsekvens måske blive syge. Derudover kan det være at udluftning i sig selv ikke er nok til at kunne nedbringe CO₂ koncentrationen i alle klasseværelser.

Tidsrammer

Projektet vil starte den 6. marts 2023 og vil blive afsluttet den 14. april 2023.

Væsentlige risici

Produktet kan være mere til besvær end gavn. Her tænkes på, hvis automatiseringen ikke kommer til at fungere.

Folkeskolelærer kan være imod mere teknologi i daglig dagen, specielt ældre kan være mere konservative, når det kommer til innovation og IT.

Bilag 3: Raspberry PI – log_MQ135.py

<https://github.com/Magn-Broder/IoT-Projekt-2>

31.03.2023 21.12

log_MQ135.py *

```

1  #!/home/magn3442/Project/IoT2 python3
2  import paho.mqtt.client as mqtt
3  import sqlite3
4  import time
5
6  db_name = 'sensor_data.db'
7
8  # log sensor data on database
9  def log_data(ppms):
10
11     conn=sqlite3.connect(db_name)
12     curs=conn.cursor()
13     curs.execute("INSERT INTO MQ135_data values(datetime('now'), (?))", (ppms,))
14     conn.commit()
15     conn.close()
16
17     def on_connect(client, userdata, flags, rc):
18         print('Connected with result code {0}'.format(rc))
19         # Subscribe (or renew if reconnect).
20         client.subscribe('CO2_ppm')
21
22     # Callback fires when a published message is received.
23     def on_message(client, userdata, msg):
24         # Decode ppm of co2 values from binary message payload.
25         ppm = [float(x) for x in msg.payload.decode("utf-8").split(',')]
26         # print('{0} ppm'.format(ppm))
27         ppms = ppm[0]
28         ppm_int = int(ppms)
29         # print(ppm_int)
30         log_data(ppm_int)
31
32     client = mqtt.Client()
33     client.on_connect = on_connect # Specify on_connect callback
34     client.on_message = on_message # Specify on_message callback
35     client.connect('localhost', 1883, 60) # Connect to MQTT broker (also running on Pi).
36
37     # Processes MQTT network traffic, callbacks and reconnections. (Blocking)
38     client.loop_forever()

```

Bilag 4: Raspberry PI – MQ135_web.py

<https://github.com/Magn-Broder/IoT-Projekt-2>

31.03.2023 11.20

MQTT_web.py

```

1 #!/home/magn3442/Project/IoT2 python
2 from matplotlib.backends.backend_agg import FigureCanvasAgg as FigureCanvas
3 from matplotlib.figure import Figure
4 import io
5
6 from flask import Flask, render_template, send_file, make_response, request
7 app = Flask(__name__)
8
9 import sqlite3
10 conn=sqlite3.connect('../sensor_data.db', check_same_thread=False)
11 curs=conn.cursor()
12
13 # Modtag den nyeste data fra databasen
14 def get_last_data():
15     for row in curs.execute("SELECT * FROM MQ135_data ORDER BY timestamp DESC LIMIT 1"):
16         time = row[0]
17         ppm = row[1]
18         print(time)
19     #conn.close()
20     return time, ppm
21
22 def get_hist_data (num_samples):
23     curs.execute("SELECT * FROM MQ135_data ORDER BY timestamp DESC LIMIT "+str(num_samples))
24     data = curs.fetchall()
25     dates = []
26     ppm = []
27
28     for row in reversed(data):
29         dates.append(row[0])
30         ppm.append(row[1])
31
32     return dates, ppm
33
34 def max_rows_table():
35     for row in curs.execute("select COUNT(ppm) from MQ135_data"):
36         max_number_rows=row[0]
37     return max_number_rows
38
39 # define and initialize global variables
40 global num_samples
41 num_samples = max_rows_table()
42 if (num_samples > 101):
43     num_samples = 100
44
45 # main route
46 @app.route("/")
47 def index():
48     time, ppm = get_last_data()
49     templateData = {
50         'time' : time,
51         'ppm' : ppm,
52         'num_samples' : num_samples
53     }
54     return render_template('index.html', **templateData)
55
56 @app.route('/', methods=['POST'])
57 def my_form_post():
58     global num_samples
59     num_samples = int (request.form['num_samples'])
60     num_max_samples = max_rows_table()
61     if (num_samples > num_max_samples):
62         num_samples = (num_max_samples-1)
63     time, ppm = get_last_data()
64     templateData = {
65         'time' : time,
66         'ppm' : ppm,
67         'num_samples' : num_samples
68     }

```

```

69     return render_template('index.html', **templateData)
70
71 @app.route('/plot/ppm')
72 def plot_ppm():
73     time, ppms = get_hist_data(num_samples)
74     ys = ppms
75     fig = Figure()
76     axis = fig.add_subplot(1, 1, 1)
77     axis.set_title("CO2 [PPM]")
78     axis.set_xlabel("Samples")
79     axis.grid(True)
80     xs = range(num_samples)
81     axis.plot(xs, ys)
82     canvas = FigureCanvas(fig)
83     output = io.BytesIO()
84     canvas.print_png(output)
85     response = make_response(output.getvalue())
86     response.mimetype = 'image/png'
87     return response
88
89 if __name__ == "__main__":
90     app.run(host='0.0.0.0', port=80, debug=False)

```

Bilag 5: ESP32 Nr.1 – boot.py

<https://github.com/Magn-Broder/IoT-Projekt-2>

```

31.03.2023 15.05                               [boot.py] *

1 # This file is executed on every boot (including wake-boot from deepsleep)
2 #import esp
3 #esp.osdebug(None)
4 #import webrepl
5 #webrepl.start()
6
7 from time import sleep
8 import network
9
10 def do_connect():
11
12     sta_if = network.WLAN(network.STA_IF)
13     if not sta_if.isconnected():
14         print('connecting to network...')
15         sta_if.active(True)
16         sta_if.connect('MSI7660', 'Kdmmmy5c#')
17         while not sta_if.isconnected():
18             pass
19     print('network config:', sta_if.ifconfig())
20
21 sleep(60)
22 do_connect()

```

Bilag 6: ESP32 Nr.1 – main.py

<https://github.com/Magn-Broder/IoT-Projekt-2>

31.03.2023 11.14 [main.py]

```

1  from umqtt.simple import MQTTClient
2  from machine import I2C, Pin
3  import _thread as thread
4  from time import sleep
5  import neopixel
6
7
8  # MQTT variabler
9  SERVER = 'raspberrypi.local' # MQTT Server Addresse
10 CLIENT_ID = 'ESP32_MQ135_Sensor'
11 TOPIC = b'C02_ppm'
12 client = MQTTClient(CLIENT_ID, SERVER)
13
14 # Neopixel variabler
15 n = 12
16 p = 15
17 np = neopixel.NeoPixel(Pin(p), n)
18
19 # Neopixel funktioner
20 def np_off():
21     for i in range(12):
22         np[i] = (0, 0, 0)
23         np.write()
24
25 def np_Green():
26     for i in range(12):
27         np[i] = (0, 50, 0)
28         np.write()
29
30 def np_Yellow():
31     for i in range(12):
32         np[i] = (50, 50, 0)
33         np.write()
34
35 def np_Red():
36     for i in range(12):
37         np[i] = (50, 0, 0)
38         np.write()
39
40 # Initialize I2C bus
41 i2c = I2C(scl=Pin(22), sda=Pin(21), freq=400000)
42 address = 0x49
43
44 # Function to read ADC data from MCP3021
45 def read_adc():
46     data = bytearray(2)
47     i2c.writeto(address, bytes([0x68]))
48     i2c.readfrom_into(address, data)
49     adc = ((data[0] & 0x0F) << 8) | data[1]
50     return adc
51
52 # Function to convert ADC data to CO2 concentration in ppm
53 def adc_to_ppm(adc):
54     ADC_0 = 300 # Replace with your sensor's values
55     ADC_400 = 1700
56     ADC_10000 = 4095
57     ppm_400 = 400
58     ppm_10000 = 10000
59     ppm = round((adc - ADC_0) * (ppm_10000 - ppm_400) / (ADC_10000 - ADC_400) + ppm_400)
60     return ppm
61
62 # Funktion til tråd 1
63 def sensor_indikator():
64     while True:
65         adc_value = read_adc()
66         co2_ppm = adc_to_ppm(adc_value)
67         print(f"C02 Level: {co2_ppm} ppm")
68         sleep(1)

```

```

69         if co2_ppm <= 500:
70             np_Green()
71
72         elif co2_ppm <= 800:
73             np_Yellow()
74
75     else:
76         np_Red()
77
78 # Funktion til tråd 2
79 def mqtt_publish_til_RPI():
80     while True:
81         adc_value = read_adc()
82         co2_ppm = adc_to_ppm(adc_value)
83         msg = (b'{}'.format(co2_ppm))
84         client.publish(TOPIC, msg) # Publish sensor data to MQTT topic
85         print(msg)
86         sleep(30)
87
88 # Opret forbindelse til MQTT broker
89 client.connect()
90
91 # Sluk Neopixel
92 np_off()
93
94 # Start 2 nye tråde
95 thread.start_new_thread(sensor_indikator,())
96 thread.start_new_thread(mqtt_publish_til_RPI,())
97

```

Bilag 7: BOM Liste

	Mængde	Navn	Beskrivelse	Pris	Fabrikant	Datasheet
	2	ESP32-WROOM-32E	Micro Controller	45 kr.	Espressif	Link
	1	Raspberry pi 3 A+	Lille Computer	249 kr.	Raspberry Pi	Link
	1	Servo motor	Motor med et potentiometer	52,50 kr.		Link
	1	MQ135 Luftkvalitet sensor	Sensor til at måle luftkvalitet og CO2	49 kr.		Link
	1	Micro USB Breakout module	Modul til at forsyne flere kilder.	7,50 kr.		Link
	1	NeoPixel Ring	LED ring, der kan lyse RGB.	49 kr.		Link
	1	MCP3021	ADC			Link

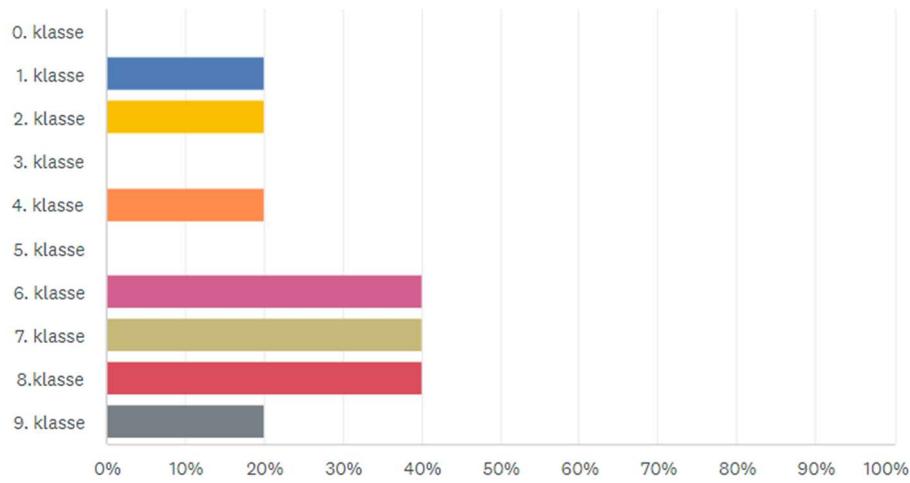
Bilag 8: Spørgeskema – CO2 i klasselokaler.

Se hele spørge skema samt yderligere information ved at klikke på linket:

https://da.surveymonkey.com/stories/SM-o409_2BTd6s2LtV9fYt4NTvw_3D_3D/

Hvilket klassesetrin er du lærer for?

Besvaret: 5 Sprunget over: 0

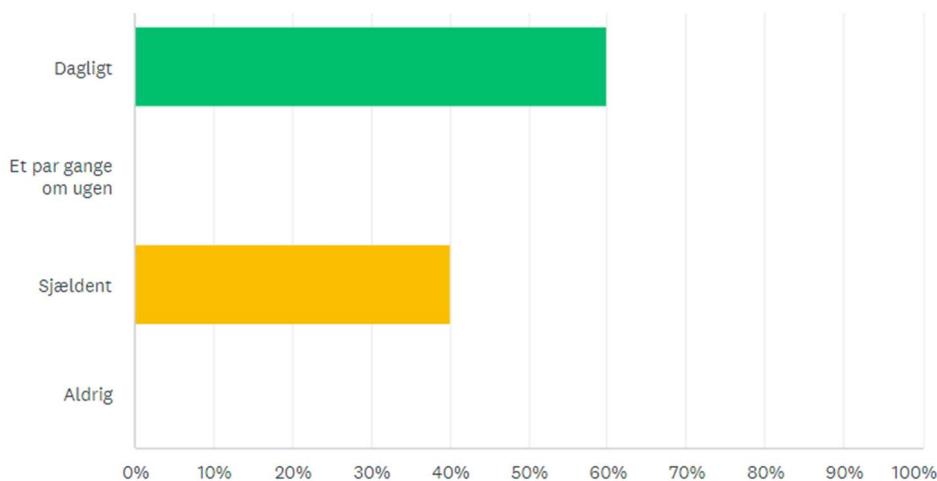


CO2 i klasselokaler.

(0)

Hvor tit gør du dig tanket om indeklimaet i dit klasselokale?

Besvaret: 5 Sprunget over: 0

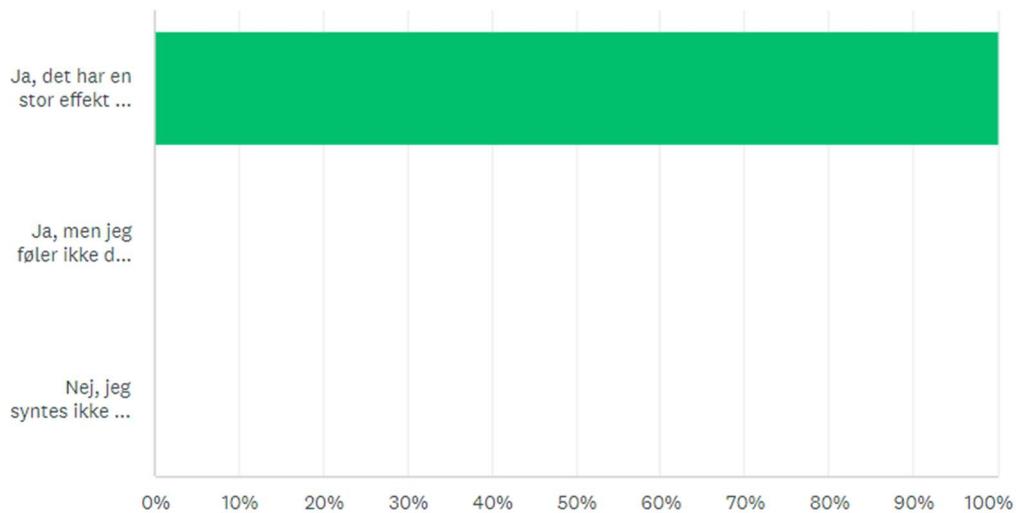


CO2 i klasselokaler.

(0)

Syntes du at dårligt indeklima er et stort problem i klasselokalet?

Besvaret: 4 Sprunget over: 1

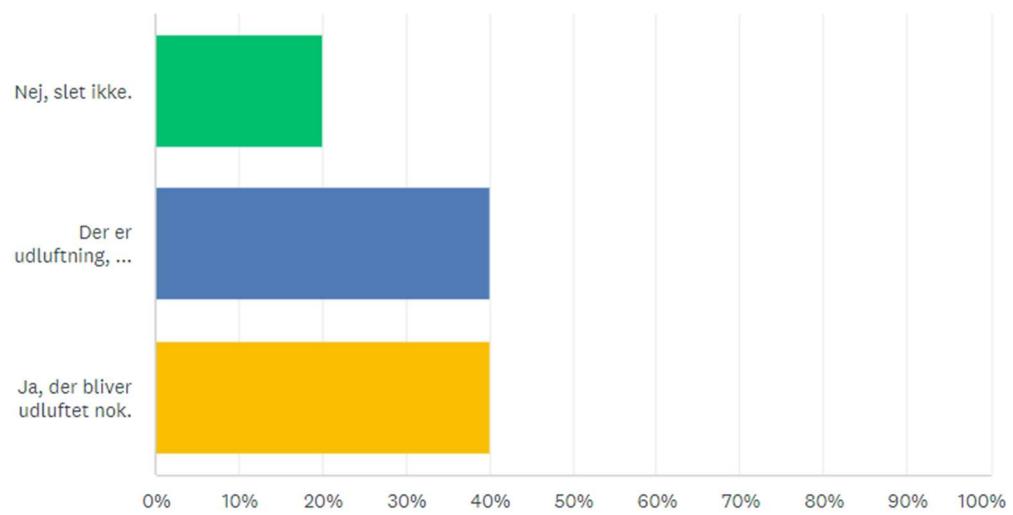


CO2 i klasselokaler.

▼ (0)

Syntes du der er tilstrækkelig udluftning i klasselokalet?

Besvaret: 5 Sprunget over: 0

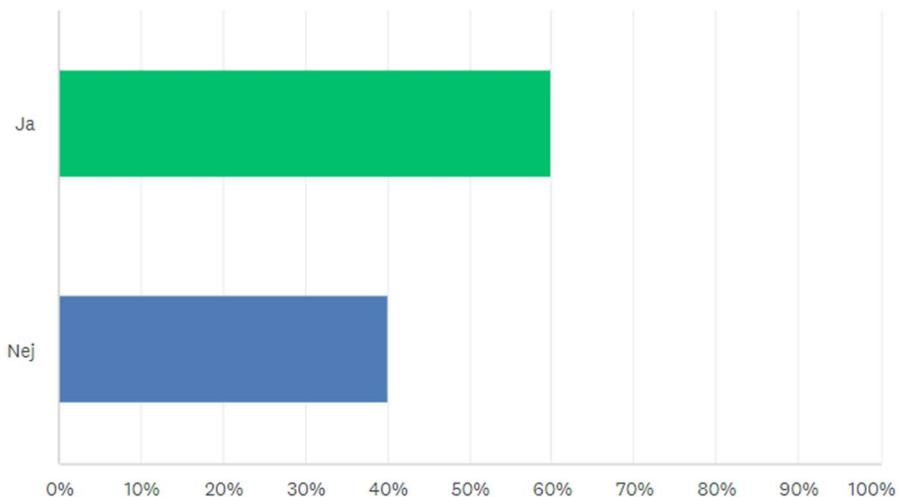


CO2 i klasselokaler.

▼ (0)

Er du bekendt med nogle konsekvenserne af for høj CO2 i klasselokalet?

Besvaret: 5 Sprunget over: 0

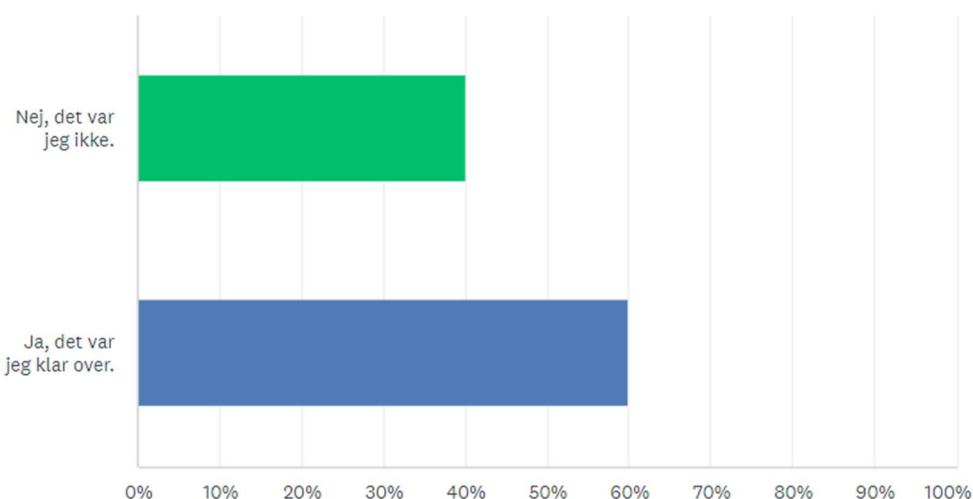


CO2 i klasselokaler.

Y (0)

Vidste du at grundskole klasselokaler typisk er overbelastet ift. CO2 n... ↗

Besvaret: 5 Sprunget over: 0

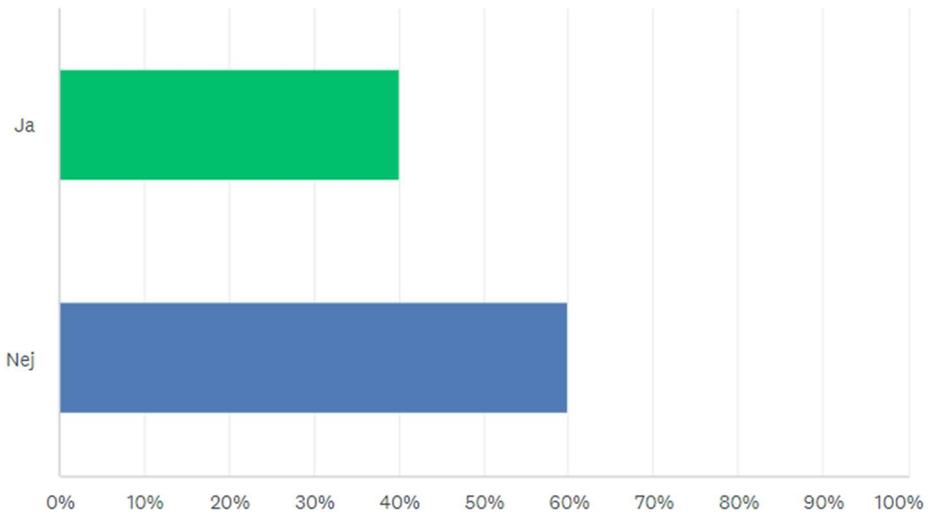


CO2 i klasselokaler.

Y (0)

Har du overvejet et gøre noget ved indeklimaet i dit klasselokale den s...

Besvaret: 5 Sprunget over: 0

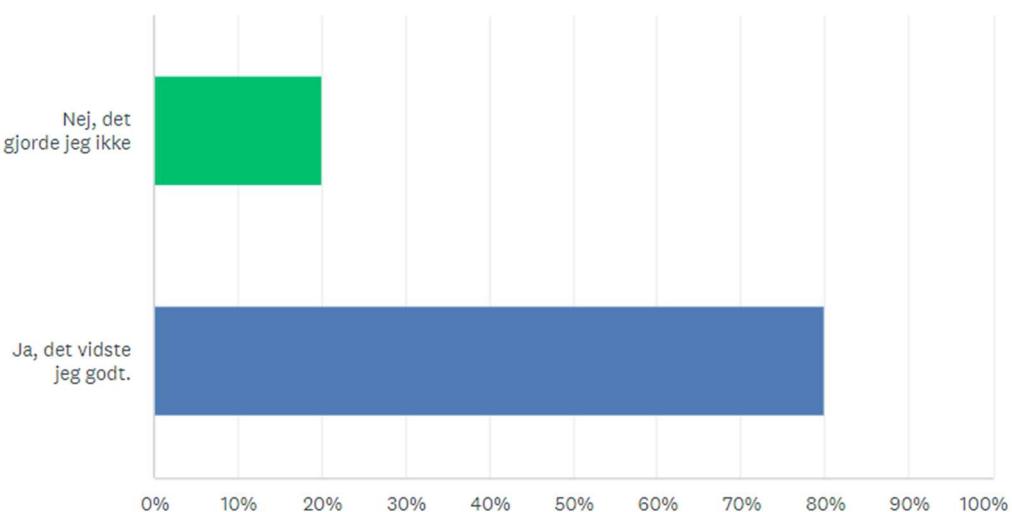


CO2 i klasselokaler.

▼ (0)

Vidste du at CO2 har en indvirkning på elevernes koncentration?

Besvaret: 5 Sprunget over: 0

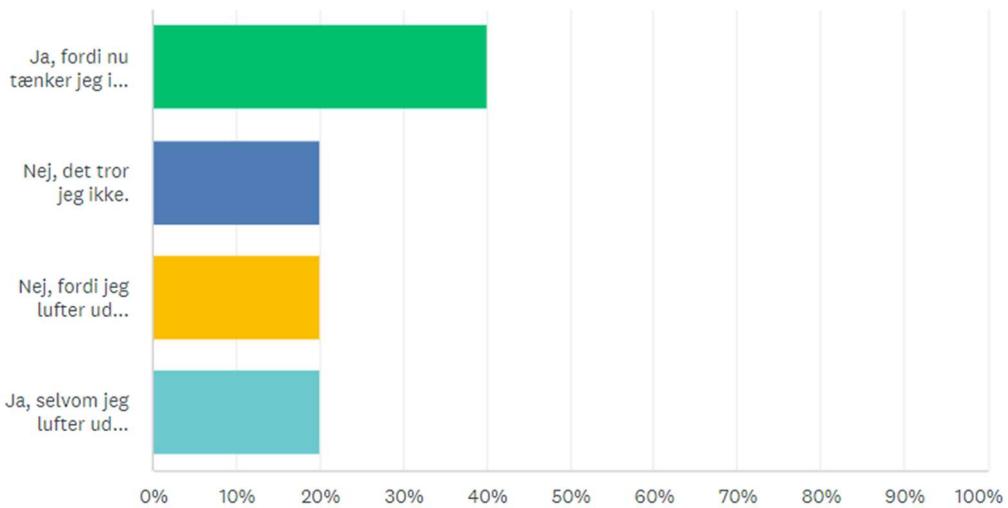


CO2 i klasselokaler.

▼ (0)

Ville detøre det nemmere for dig at holde et godt indeklima i klassel...

Besvaret: 5 Sprunget over: 0

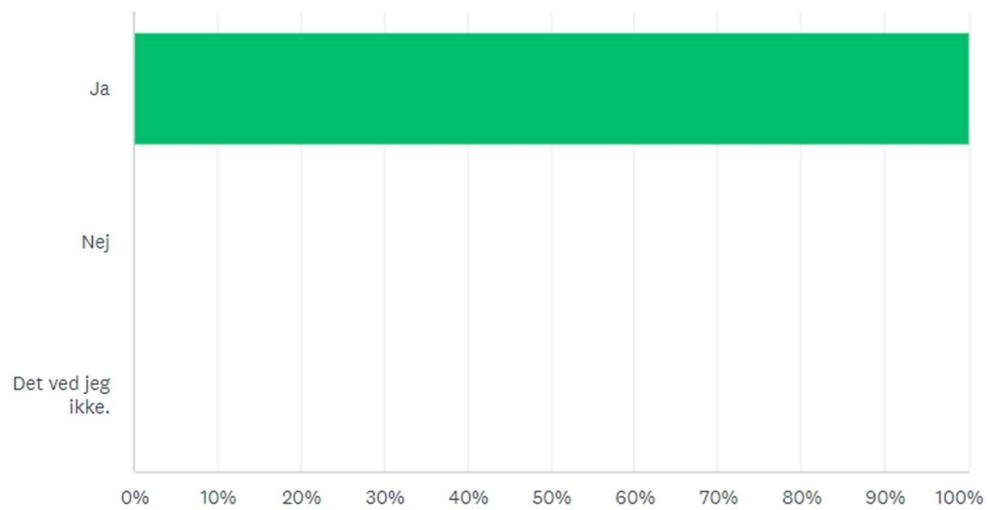


CO2 i klasselokaler.

▼ (0)

Ville det hjælpe at automatisere udluftning, i forhold til det anbefaled...

Besvaret: 5 Sprunget over: 0



CO2 i klasselokaler.

▼ (0)

Bilag 9: Skærm billeder af hjemmeside med 3 timers mellemrum

