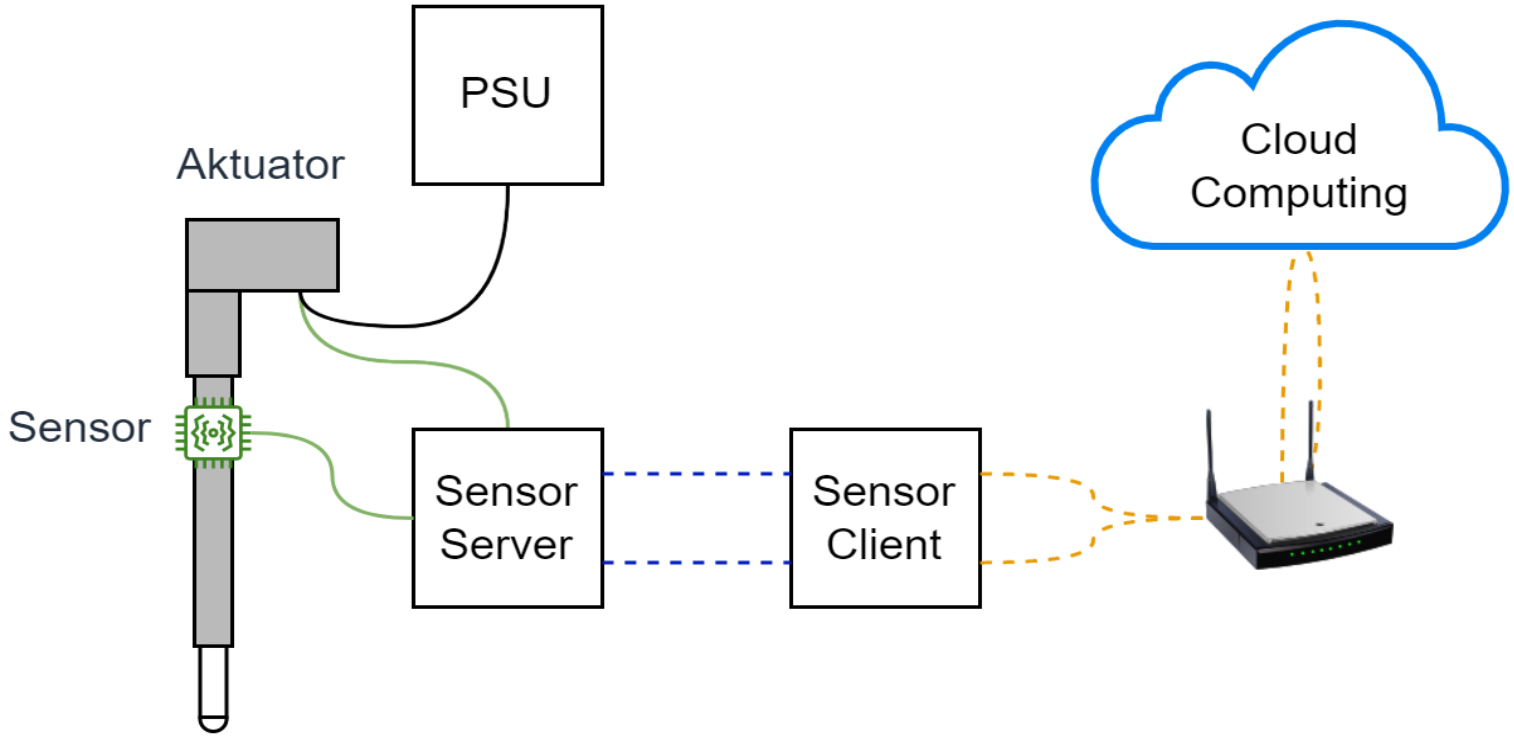


IIoT prædiktiv vedligeholdelsessystem

Elektronik Diplomingeniør Afgangsprojekt

Elektronik og IT, 01-2023

P7 Afgangsprojekt

STUDENTER RAPPORT
AALBORG UNIVERSITY

Copyright © Aalborg University 2021

AALBORG UNIVERSITET
STUDENTERRAPPORT

Elektronik og IT
Aalborg Universitet
https://www.aau.dk

Titel:
IIoT prædiktiv vedligeholdelsessystem

Tema:
Elektronik Diplomingeniør
Afgangsprojekt

Projektperiode:
P7 2022

Deltager(e):
Frederik Bang

Vejleder(e):
Gilberto Berardinelli

Oplagstal: Tilgængelig online

Sidetal: 125

Afleveringsdato:
05-01-2023

Abstract:

Denne rapport omhandler udviklingen af et prototype IIoT prædiktiv vedligeholdelsessystem bestående af en Sensor Server, en Sensor Client og AWS Cloud Computing. Sensor Server måler vibrationsdata for en elektrisk lineær aktuator med accelerometret i en IMU, når der modtages en anmodning om sensordata fra Sensor Clienten sendes de over BLE Mesh. BLE Mesh datatransmissionen er delt op i fire datatransmissioner, idet datamængden overskrider den maksimale for én transmission. Sensor Clienten er tilsluttet eksisterende Wi-Fi, og sender over MQTT nye vibrationsdata til AWS. Ved modtagelse af nye vibrationsdata i AWS anvendes "Backend" Lambda til udførelse af en simpel CbM, hvor amplituderne ved de forskellige harmoniske frekvenser sammenlignes med værdier for en velfungerende aktuator. Ved målinger uden for specifikation sendes en Fejl Indikations E-mail indeholdende enhedsnavn, og hvilke harmoniske frekvender, der var uden for specifikation til en serviceperson. Prototype IIoT prædiktiv vedligeholdelsessystemet overholder 4/4 funktionelle krav og 1/1 tekniske krav.

Rapportens indhold er frit tilgængeligt, men offentliggørelse (med kildeangivelse) må kun ske efter aftale med forfatterne.

Frederik Bang
<fbang18@student.aau.dk >

Forord

Denne rapport er udarbejdet som Elektronik Diplomingeniør Afgangsprojekt under 7. semester i efteråret 2022 på *Elektronik og IT* uddannelsen.

Læsevejledning

I rapporten vil der fremtræde kildehenvisninger og en samlet kildeliste over disse kan findes bagerst i rapporten inden bilag og appendiks. Kildehenvisningen er efter Vancouvermetoden, så i teksten refereres en kilde til på formen "[Nr.]", hvor kilderne er nummereret efter kronologisk optrædende. Ligeledes er kildelisten nummereret efter optræden. Hvis årstallet for udgivelsen var utilgængelig, vil kildelisten vise årstallet for besøgelse. Henvisningerne fører med hyperlinks ned til kildelisten, hvor kilder er angivet med forfatter, titel, evt. link, udgivelsesmåned, -år og dato for besøg. Figurer, tabeller mm. er nummereret i forhold til hvilket kapitel de findes i, hvilket vil sige, at den første figur i kapitel to har nummer 2.1 og den næste har 2.2 osv. Forklarende tekst for figurer findes under de givne figurer, hvorimod den forklarende tekst til tabeller findes over. Efter kildelisten findes appendiksene og bilagene, der bliver nummeret med bogstaver.

Fodnoten bliver brugt som *den forklarende/uddybende note*, hvori en forklaring af enkelte ord/forkortelser indgår.

Der benyttes videnskabelige præfikser og enheder i overensstemmelse med ISO-80000.

Fordi symbolet for en decimalseperator er forskelligt i Danmark, og i det programmeringssprog der bruges til at udarbejde produktet, gøres der opmærksom på, at programkode på Github vil bruge et punktum (.) som decimalseperator, mens diverse tal og udregninger vil gøre brug af et komma (,).

Akronymliste er vist på følgende side, hvor der er henvisninger til de sider, hvor akronymet anvendes. Akronymer skrives ud i fuld form første gang de anvendes fra side 1.

Akronymer

AWS Amazon Web Services. i, viii, 16–18, 20, 22, 23, 27, 38, 39, 44, 45, 47, 58, 61–64, 68, 70, 71, 73–76, 101, 106, 114, 121–125

BaaS Backend as a Service. 16, 17, 20, 44, 74

BLE Bluetooth Low Energy. i, vii, viii, 12–14, 18–20, 22, 25, 28–35, 37, 45, 47–50, 54–63, 67–69, 72–75, 77–79, 87, 88, 109, 114, 115

CbM Condition-based Monitoring. i, vii, 6, 7, 19, 21, 23, 27, 37, 38, 41–45, 64, 65, 73, 74, 76–79, 101, 103, 105, 107

CHIP Connected Home over IP. 15

ESP Espressif. 28, 30, 31, 38, 46–48, 51, 52, 54, 55, 57, 59, 61, 63, 75, 87, 88, 97, 98, 102, 110, 117

FFT Fast Fourier Transform. 7, 38, 41, 42, 44, 64–66, 68, 70, 71, 74, 76, 101, 106, 114, 121–125

HTTP Hypertext Transfer Protocol. 18

IaaS Infrastructure as a Service. 16, 17, 20

IDF IoT Development Framework. 48

IIoT Industrial Internet of Things. i, vii, 1–3, 5, 9–13, 15, 17–23, 25, 27, 28, 30, 37–39, 45–48, 67, 69, 71–73, 77, 101, 109, 115, 117, 125

IMU Inertial Measurement Unit. i, 28, 48, 75, 77

IoT Internet of Things. 1, 15, 38, 39, 44, 45, 47, 62, 64, 65, 74, 76

JSON JavaScript Object Notation. 40

JTAG Joint Test Action Group. 47, 48

LPWAN Low Power Wide Area Networks. 11, 14, 20

MQTT Message Queuing Telemetry Transport. i, vii, viii, 18, 27, 38–40, 44, 45, 58, 59, 61–65, 74–76, 79

NTP Network Time Protocol. 38, 61, 63, 74, 75

PaaS Platform as a Service. 16, 17, 20

PDU Protocol Data Unit. 31–37, 74, 77, 78

PSU Power Supply Unit. 7, 19, 68, 70, 91–93, 102, 104, 106, 110, 112, 114, 117, 119, 121

RFID RadioFrekvens IDentifikation. 11

RPM Revolutions Per Minute. 7, 8, 19, 25, 42, 43, 74, 91, 101

RTD Resistance Temperature Detector. 77

RTOS Real-Time Operating System. viii, 45, 47–50, 57–59, 61, 63, 75

S3 Simple Storage Service. 38, 44, 45, 64–66, 74, 76

SaaS Software as a Service. 16, 17, 20

SNS Simple Notification Service. 38, 44, 45, 64, 65, 74, 76

SoC System on a Chip. 47, 48, 57, 63

TLS Transport Layer Security. 18, 20

VC Code Visual Studio Code. 48, 100

WPAN Wireless Personal Area Network. 11

Indholdsfortegnelse

Kapitel 1 Introduktion **1**
1.1 Initierende problemformulering 3

Kapitel 2 Analyse **5**
2.1 Motivation 5
2.1.1 Økonomisk 5
2.1.2 FN's verdensmål 6
2.2 Prædiktiv vedligeholdelse af elektrisk lineær aktuatore 6
2.2.1 Nøgleparametre 6
2.2.2 Datamængde 7
2.3 Kommunikation 9
2.3.1 Trådet og trådløse forbindelser 10
2.3.2 Trådløse IIoT kommunikationsteknologier 11
2.4 Cloud Computing 15
2.4.1 Udbydere 16
2.5 Cyber Security 18
2.6 Delkonklusion 19
2.6.1 Ideelt system 21
2.7 Afgrænsning 22
2.7.1 Problemformulering 23

Kapitel 3 Kravspecifikation **25**

Kapitel 4 Design **27**
4.1 Dataopsamling 28
4.1.1 Vibrationsmåling 28
4.2 BLE Mesh 30
4.2.1 Baggrundsinformation 31
4.2.2 Design af dataopsplitning 34
4.3 Cloud Computing 38
4.3.1 Wi-Fi 38
4.3.2 MQTT 39
4.3.3 CbM 41

4.3.4 AWS Cloud Computing 44

Kapitel 5 Implementering **45**
5.1 Fysisk prototype 46
5.1.1 Hardware 47
5.1.2 Udviklingsmiljø 48
5.2 Sensor Server 49
5.2.1 Datahåndtering 49
5.2.2 Dataopsamling 51
5.2.3 BLE Mesh 54
5.2.4 RTOS 57
5.3 Sensor Client 58
5.3.1 Datahåndtering 58
5.3.2 BLE Mesh 59
5.3.3 Wi-Fi 61
5.3.4 AWS MQTT 61
5.3.5 RTOS 63
5.4 Cloud Computing 64

Kapitel 6 Accepttest **67**
6.1 Funktionelle krav 67
6.1.1 Fremgangsmåde 68
6.1.2 Konklusion 69
6.2 Tekniske krav 70
6.2.1 Fremgangsmåde 70
6.2.2 Konklusion 71
6.3 Konklusion af accepttest 72

Kapitel 7 Konklusion **73**

Kapitel 8 Diskussion **77**

Litteratur **81**

Appendiks **85**

Appendiks I ESP BLE Mesh Arkitektur **87**

Målejournal **89**

Målejournal A Elektrisk lineær aktuator harmoniske frekvenser **91**

Målejournal B Vibrationssensor Reaktionstid **97**

Målejournal C CbM Grænseværdier **101**

Målejournal D Accepttest af funktionelle krav **109**

Målejournal E Accepttest af tekniske krav **117**

Introduktion 1

Internet of Things (IoT) er sammenkoblingen af fysiske enheder med internettet til opsamling og udveksling af data. De opsamlede data kan f.eks. sendes videre til lagring eller databehandling i skyen, hvor det er muligt at handle på de opsamlede data. IoT integreres i alt fra belysning- og sensorenheder rettet imod privatpersoner til industrielle produktions overvågningssystemer med henblik på prædiktiv vedligeholdelse [1]. En af de store salgspunkter for IoT produkter er at automatiserer manuelle opgaver, og derved spare tid eller penge. Ved integration i industrielle systemer omtales det Industrial Internet of Things (IIoT).

For at sikre følsomme data er Cyber Security et vigtigt emne når det kommer til IIoT systemer. Cyber Security er ekstra vigtig for industrielle systemer, hvor et brud på sikkerheden kan koste mange penge og i værste tilfælde forvolde så meget skade at firmaet går konkurs. For industrielle IIoT systemer er robustheden af systemet vigtigt, idet et nedbrug på f.eks. en fremstillingsfabrik kan ligesom sikkerhedsbrug være meget skadende for firmaet.

I industrien er det vigtigt at nye systemer er nemme at implementere så antal mandetimer og nødvendige produktionsstop under opsætningen mindskes. Opsætningens forbrug af mandetimer og nødvendige produktionsstop vil være afhængig af den valgte kommunikations type og protokol.

I fremstillingsfabrikker foretages der ofte forebyggende vedligeholdelse, hvor sliddele udskiftes efter de har været brugt et bestemt antal gange eller i en bestemt tid. Ud fra en række tests eller statistik for produktionsudstyret kan det udledes, hvornår sliddelen burde udskiftes. Dette udskiftningspunkt sættes generelt så sliddelen udskiftes før den dårligste ydenede sliddelen fra testen nedbryder. Derudover kan det i vitale udstyr være nødvendig med en ekstra margen på udskiftningspunktet for at mindske risikoen for utilsigtet nedbrug [2].

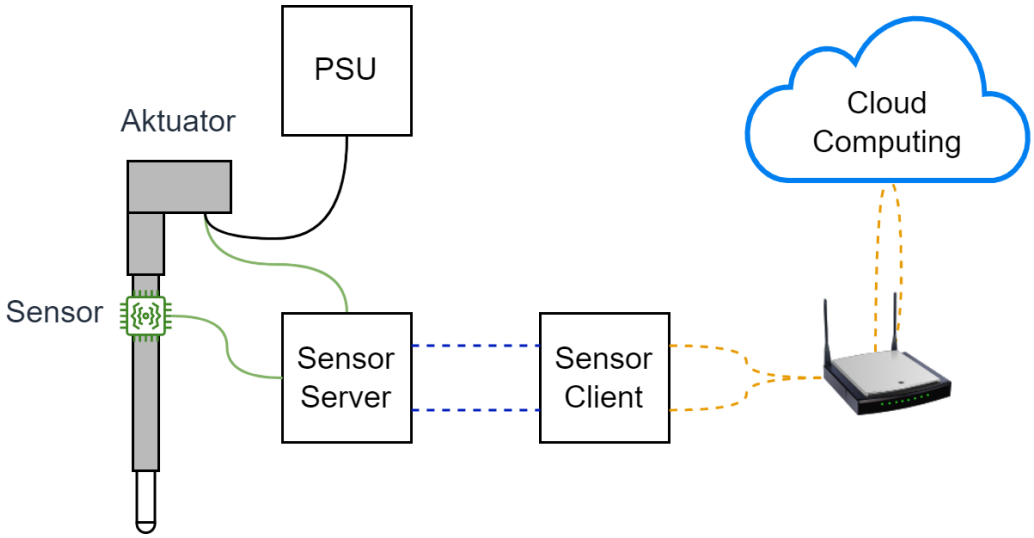
Dette resulterer i at der udskiftes en række potentielt venfungerende sliddele, der forsat kunne have været brugt yderligere i en periode uden nedbrud. Samtidig laves der potentielt flere servicestop af produktionsudstyrret end der er nødvendigt.

En løsning på dette kan være *prædiktiv vedligeholdelse*.

Et prædiktiv vedligeholdelsessystem overvåger en række nøgleparametre for sliddelen, og sammenligner dem med kendte værdier for en fungerende enhed. Ud fra dataene kan systemet bestemme, hvornår sliddelen skal udskiftes. Dette sørger for at sliddelen først udskiftes når det er nødvendigt, hvilket mindsker udskiftningen af potentielt velfungerende sliddele [2].

En gængs brugt sliddel er elektriske lineær aktuatorer, der bl.a. anvendes i mange fremstillingsfabrikker. Et IIoT prædiktiv vedligeholdelsessystem vil kunne anvendes til at overvåge nøgleparametre for den elektriske aktuator, hvorefter lagring og databehandling kan gøres i skyen.

Figur 1.1 illustrere et IIoT prædiktiv vedligeholdelsessystem, hvor en aktuator overvåges med en sensor tilsluttet en Sensor Server, der sender dataene over et lokalt netværk til en Sensor Client, der videresender dataene gennem en internetforbindelse til en Cloud Computing ydelse.



Figur 1.1: IIoT prædiktiv vedligeholdelsessystem system.

Projektet er udarbejdet i samarbejde med Danotek A/S, hvor deres R&D Ingeniøre har været anvendt som sparingspartnere under udviklingen af prototype IIoT prædiktiv vedligeholdelsessystemet.

1.1 Initierende problemformulering

Hvad er motivationen for implementering af et IIoT prædiktiv vedligeholdelsessystem, hvilke kommunikationsteknologier kan anvendes, og hvilke overvejelser er der til Cyber Security?

Følgende skal undersøges:

- Hvilke fordele giver prædiktiv vedligeholdelse?
  - Evt. sammenhæng med FN's verdensmål
- Hvordan kan prædiktiv vedligeholdelse til elektriske lineær aktuatore implementeres?
  - Hvilke nøgleparametre er relevante at måle?
- Hvilke kommunikationsteknologi er mest anvendeligt for et IIoT prædiktiv vedligeholdelsessystem?
- Hvilke overvejelser er der til Cyber Security for IIoT systemer i industrien?

Analyse 2

Følgende kapitel omhandler analyse af et IIoT prædiktiv vedligeholdelsessystem. Der undersøges motivationen for implementering, hvordan et prædiktiv vedligeholdelsessystem af lineære aktuatore kan implementeres, hvilke kommunikationsteknologier, der er mest anvendelige, og hvilke overvejelser der er til Cyber Security.

2.1 Motivation

Motivationen for brug af prædiktiv vedligeholdelse er primært økonomisk og herefter nedsat ressourcespild.

2.1.1 Økonomisk

Opsætning af et nyt prædiktivt vedligeholdelsessystem vil indebære en opstarts udgift. De økonomiske besparelser skal derfor kunne tilbagebetale opstarts udgiften inden for et acceptabel antal måneder eller år. Den acceptable tilbagebetalings tid kan varier meget mellem firmaer.

Prædiktivt vedligeholdelse har størst indflydelse på produktionsudstyr, der afhænger af ét enkelt sliddel, hvor fejl herpå vil resultere i et langt utilsigtet nedbrud. Ved at overvåge nøgleparametre for sliddelen nedsættes risikoen for utilsigtet nedbrud på produktionsudstyret, og den udskiftes ikke før det er nødvendigt. Herudover kan reparationstiden nedsættes ved at skadelige problemer bliver opdaget tidligt, og derfor ikke forvolder skade på resten af produktionsudstyret [2].

Hver gang en produktion stoppes koster det virksomheden mange penge, hvor den tabte indtjening er afhænger af længden på nedbruddet. Ved nedsætningen af risiko for og omfang af nedbrud kan prisen for nedbrudsforsikring falde med op til 25% [2].

Ved overvågning af produktionsudstyret kan det sikres at det opfører sig som forventet, og derfor producerer produkter med samme høje standard som ønsket. Herved nedsættes risikoen for at miste en eventuel ISO-9000 certifikation [2].

2.1.2 FN's verdensmål

De 17 verdensmål beskriver, hvordan FN ønsker at gøre verden til et mere bæredygtig og bedre sted. Prædiktivt vedligeholdelse omhandler mest mål 8 vedrørende *Anstændige Jobs og Økonomisk Vækst* ved at optimerer produktionsprocessen og nedsætte løbende omkostninger vedrørende service og reparation. Ved at formindske mængden af velfungerende sliddele, der udskiftes omfattes mål 12 vedrørende *Ansvarligt Forbrug og Produktion* [3].

2.2 Prædiktiv vedligeholdelse af elektrisk lineær aktuatore

En tilgang til prædiktiv vedligeholdelse er Condition-based Monitoring (CbM), hvor en række nøgleparametre overvåges. Med f.eks. Machine Learning trænes en algoritme til at sammenligne de målte værdier med kendte værdier for en velfungerende komponent. Herved reageres der, hvis de overvågede data kommer for langt fra den kendte værdi, hvilket er et tegn på at der skal ydes service.

Ved CbM er det vigtig at analysere og opdele parametre ændringer, hvorved det kan specificeres hvilken sliddel, der bør udøves service på.

Indikation på at der skal ydes service kan implementeres på mange forskellige måder alt efter fremstillingsvirksomhedens infrastruktur. Der kan f.eks. indikeres med en lampe på udstyret, en besked i allerede implementeret overvågningsprogram eller en mail til den serviceansvarlige.

Den acceptable reaktionstid for et prædiktiv vedligeholdelsessystem er afhængig af, hvilken applikation systemet implementeres i. Applikationer, der kører kontinuerligt vil have skarpere krav til reaktionstid end applikationer, der kun kører en gang i døgnet.

2.2.1 Nøgleparametre

Til udøvelse af CbM på en elektrisk lineær aktuator skal der måles en række nøgleparametre. Hver målingstype kan anvendes til at bestemme forskellige fejlscenarier, dog vil nogle fejlscenarier kunne bestemmes af flere målingstyper.

Der måles generelt flere nøgleparametre samtidig for at danne et bedre billede af sliddelens tilstand, hvor tre vigtige nøgleparametre for en elektrisk lineær aktuator er vibration, strømtræk og temperatur.

Vibration

Den mest anvendte målingstype til prædiktiv vedligeholdelse er vibrationsmåling, der kan vise en stor række fejltilstande. Vibrationsmåling er det stærkeste værktøj til prædiktiv vedligeholdelse når det anvendes korrekt. Ved vibrationsmåling af elektriske lineær aktuatore er det nødvendigt at målingen sker i forhold til den lineære bevægelse. Derfor skal der tages højde for den tid det tage den lineær aktuatoren at komme fra den ene yderposition til den anden [2].

Ved vibrationsmåling med en samplingsfrekvens på op til 30 kHz kan der findes fejl vedrørende kuglelejer, gear, fejljustering og belastningstilstand [4]. Fejl vil generelt optræde som 1. til 4. harmoniske frekvenser af den elektriske motors Revolutions Per Minute (RPM) eller RPM efter en gearing [2].

Dataene måles i tids-domænet, hvorefter det kan konverteres til frekvens-domænet ved f.eks. Fast Fourier Transform (FFT) for at simplificere databehandlingen. Ved en enkelkanals vibrationsmåling kan der ikke dannes et komplet billede af komponentets tilstand, derfor kan der med fordel anvendes en sensor, der måler i tre akser [2].

Strømtræk

Strømtrækket af en elektrisk lineær aktuator vil ofte ændre sig, hvis der er ved at udvikles fejl. Dette kan komme af f.eks. udtørret smørremiddel i gearkassen, hvilket vil øge friktionen og derved øge strømtrækket. Måling af strømtrækket gør det muligt at detektere fejl vedrørende excentricitet af roteren, motorspole, rotor arm, Power Supply Unit (PSU) og kuglelejer [4].

Temperatur

Temperaturen af en elektrisk lineær aktuator vil ændre sig i forhold til bl.a. brugsmønster og belastning. Ved måling af temperatur kan der detekteres fejl vedrørende forøget friktion, forøget belastning, overdreven start/stop og PSU [4].

2.2.2 Datamængde

Idet der kun skal måles data imens den elektriske lineære aktuator er i bevægelse kan der med fordel tages flere målinger der efterfølgende kan tages gennemsnittet af.

Et ekstremt eksempel på CbM af en elektrisk motor, hvor en rotor arm efter en gearing roterer med 30.000 RPM opstilles.

Omdrejninger per sekund (RPS) udregnes ud fra motor RPM efter gearing (RPM):

$$RPS = \frac{RPM}{\frac{s}{min}} = \frac{30 \cdot 10^3}{60} = 500\,\text{Hz} \tag{2.1}$$

Den 4. harmoniske frekvens udregnes:

$$f_4 = f_1 \cdot 4 = 500\,\text{Hz} \cdot 4 = 2\,\text{kHz} \tag{2.2}$$

Ifølge Nyquist–Shannon skal der for at måle frekvenser (f_n) op til 2 kHz samples med en minimum samplingfrekvens (f_s) på $f_\text{n} \cdot 2 = 4$ kHz når dataene konverteres til frekvensdomænet.

Ved at måle med f.eks. 6 kHz er det derfor sikret at der kan observeres frekvenser op til 2 kHz.

Antal målinger (BL) vælges som 2 opløftet til et heltal som f.eks. $2^{10} = 1024$ målinger. Herefter kan målingstiden (D) udregnes:

$$D = \frac{BL}{f_\text{s}} = \frac{1024}{6\,\text{kHz}} \approx 171\,\text{ms} \tag{2.3}$$

Frekvensopløsningen (df) udregnes:

$$df = \frac{1}{D} = \frac{1}{171\,\text{ms}} \approx 6\,\text{Hz} \tag{2.4}$$

Ved brug af f.eks. et tre-akset accelerometer med en 16 bit opløsning kan den samlede datamængde for en dataopsamling udregnes:

$$1024 \cdot 16\,\text{bit} \cdot 3\,\text{akser} \approx 49{,}15\,\text{kbit} \tag{2.5}$$

Vibrationsmåling kræver meget større datamængder i forhold til strøm- og temperaturmåling, der vil kunne repræsenteres med f.eks. 16 bit hver. Ved måling af alle tre nøgleparametre med samme hyppighed vil den samlede datamængde øges til:

$$49{,}15\,\text{kbit} + 16\,\text{bit} + 16\,\text{bit} \approx 49{,}18\,\text{kbit} \tag{2.6}$$

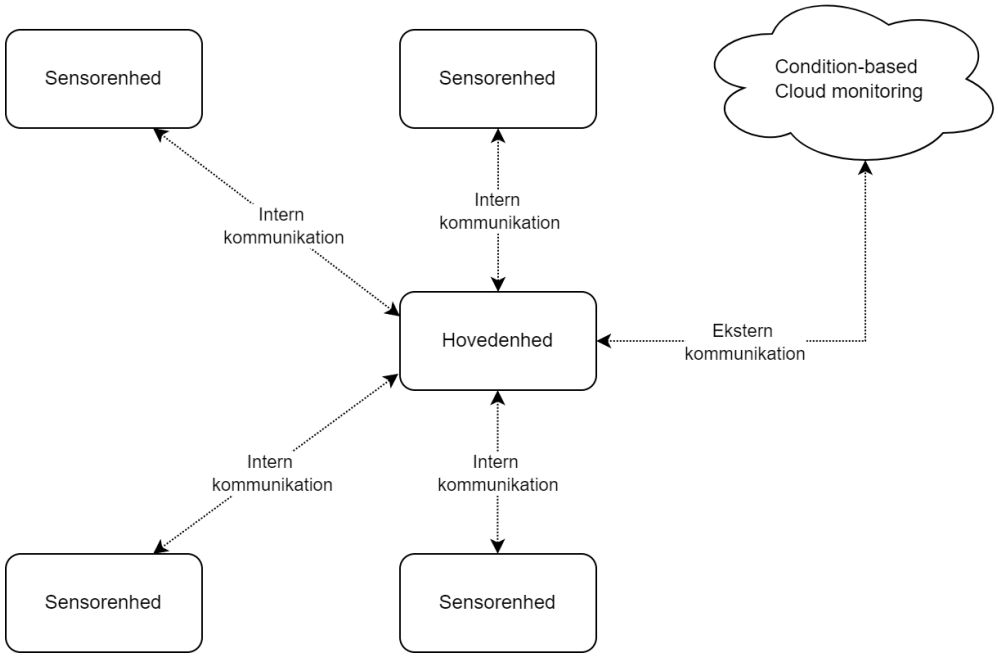
Hyppigheden af målingerne kan have indflydelse på funktionaliteten af den senere databehandling, hvor en for langsom hyppighed kan resulterer i fejl.

2.3 Kommunikation

Ved at opdele et IIoT prædiktiv vedligeholdelsessystem i intern og ekstern kommunikation sikres det at der kun er en forbindelse til internettet, som er den mest udsatte forbindelse.

Intern kommunikation er den kommunikation, der sker inden for firmaets grund og bestå af flere sensorenheder, der udveksler data med en gateway. Det simpleste netværk til intern kommunikation er et Star netværk, hvor hver sensorenhed kommunikerer direkte med hovedenheden. Et Star netværk kan implementeres med enten trådet eller trådløse forbindelser.

Ekstern kommunikation er den kommunikation, der sker fra firmaets grund og ud på internettet og omhandler dataudveksling mellem gatewayen og en Cloud Computing service, hvor gatewayen skal have en enten trådet eller trådløs forbindelse til internettet. Figur 2.1 viser et eksempel på intern Star netværk og ekstern kommunikation netværk for et IIoT prædiktiv vedligeholdesystemet.



Figur 2.1: Eksempel kommunikation netværk for et IIoT prædiktiv vedligeholdesystemet.

2.3.1 Trådet og trådløse forbindelser

Trådet forbindelser tager generelt lang tid at implementere idet der skal trækkes fysiske kabler til alle kommunikationsforbindelser. Derfor kan det være nødvendigt at lukke en produktion ned i længere tid under installation alt efter, hvordan kabelføringen skal være. Trådet forbindelser er generelt upåvirkede af omgivelserne systemet opsættes når systemet er færdig implementeret. Efterfølgende vedligeholdelse af forbindelserne vil kunne være langtrukne idet brud på fysiske kabler kan være svære at lokalisere.

Trådløse forbindelser er generelt hurtige at implementere idet der ikke skal trækkes nogle fysiske forbindelser mellem enhederne. Dette resultere i at nedlukningen af produktionen holdes til et minimum. Trådløse forbindelser er generelt påvirkede af deres omgivelser som f.eks. armerede betonvægge, der i værste tilfælde kan blokerer det trådløse signal. Efterfølgende vedligeholdelse vil være hurtigt idet en fejlende enhed nemt kan udskiftes.

Tabel 2.1 viser de generelle fordele og ulemper ved trådet og trådløse forbindelser.

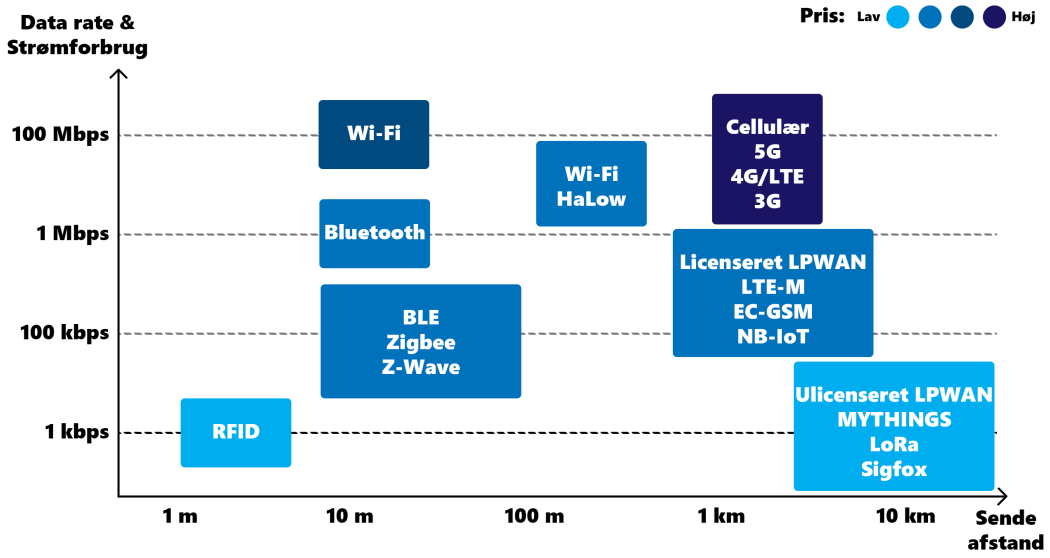
Tabel 2.1: Kommunikationstyper generelle fordele og ulemper.

Type	Fordele	Ulemper
Trådet forbindelser	Forbindelse upåvirket af omgivelser. Lav latency. Stor pålidelighed.	Langsom implementering. Besværlig vedligeholdelse.
Trådløse forbindelser	Hurtig implementering. Nem vedligeholdelse. God skalarbarhed. Monitorering af mobile objekter.	Forbindelse påvirket af omgivelser. Moderat latency.

Ved implementering af et IIoT prædiktiv vedligeholdelsessystem i en fremstillingsvirksomhed er det vigtigste at begrænse den tid, hvor produktionen er stoppet. Derfor er den mest hensigtsmæssige kommunikationstype for intern kommunikation en trådløs forbindelse.

2.3.2 Trådløse IIoT kommunikationsteknologier

Trådløse IIoT kommunikationsteknologiers vigtigste parametre omhandler data rate, strømforbrug, sendeafstand og pris. Figur 2.2 viser en sammenligning af de forskellige teknologier:



Figur 2.2: Sammenligning af trådløse IIoT kommunikationsteknologier inspireret af [5].

På figur 2.2 beskriver X-aksen den forventelige sende afstand og Y-aksen beskriver den forventelige data rate og strømforbrug, der generelt afhænger af hinanden. Farven af de forskellige kasser beskriver prisklassen af teknologien på en fire-trins skala.

Der er styrker og svagheder ved de forskellige kommunikationsteknologier, hvilket gør at der ikke kan udvælges en som generelt bedst. Derfor undersøges det hvilken, der er bedst egnet til intern kommunikation og derefter ekstern kommunikation.

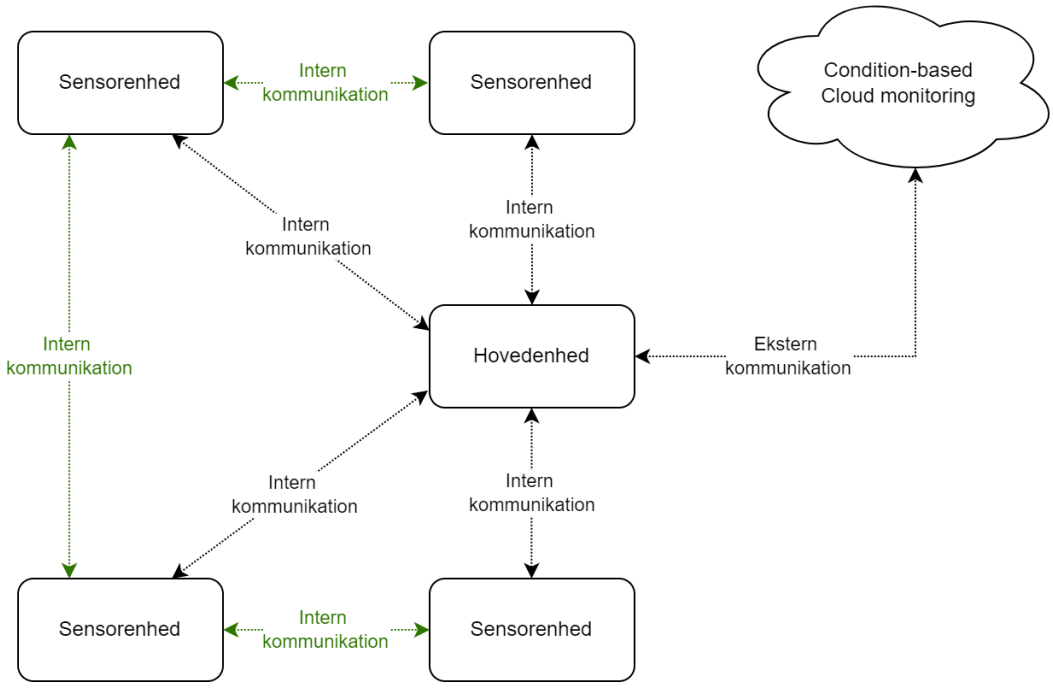
Intern kommunikation

Wi-Fi, Wi-Fi HaLow og Bluetooth kan opnå hurtige dataoverførsler dog på bekostning af større strømforbrug. RadioFrekvens IDentifikation (RFID) bruger meget lidt strøm, men har en meget begrænset hastighed og sende afstand. Low Power Wide Area Networks (LPWAN) og Cellulær kan sende over store afstande, men de er enten dyre, har begrænset sendehastig eller kan ikke opsættes som et Wireless Personal Area Network (WPAN).

Firmaer kan købe adgang til at anvende 5G til private netværk inden for firmaets adresse. Dette gør det muligt at anvende den hurtige data rate fra 5G uden at netværket kan tilgås utilsigtet udefra. Dette er dog en meget dyr løsning, der kan koste milioner i opstartsudgifter [6].

Til intern kommunikation med små afstande og små data mængder som udledt i afsnit 2.2.2, sammenlignes Bluetooth Low Energy (BLE), Zigbee og Z-Wave idet de samlet set har de mest egnede kompromiser af data rate, strømforbrug og sende afstand.

BLE, Zigbee og Z-Wave kan alle opsættes som Mesh netværk, hvilket giver en række fordele i forhold til et Star netværk. I Mesh netværk kan sensorenhederene både kommunikerer direkte med gatewayen og med hinanden for at sende beskeder videre til gatewaten. Hver gang data videretransmitteres af en sensorenhed kaldes det et "Hop". Mesh netværk kan derfor øge afstanden mellem sensorenheder og gatewayen idet der ikke behøves en direkte forbindelse. Figur 2.3 viser et eksempel på intern Mesh netværk og ekstern kommunikation netværk for et IIoT prædiktiv vedligeholdesystemet, hvor de ekstra forbindelser mellem sensorenhederne er markeret med grøn.



Figur 2.3: IIoT Mesh netværk kommunikation.

Tabel 2.2 viser de vigtigste parametre for BLE, Zigbee og Z-Wave opsat som Mesh netværk, hvor de har sammenlignelige strømforbrug.

Tabel 2.2: Sammenligning af BLE, Zigbee og Z-Wave konfigureret som Mesh netværk [7] [8] [9] [10].

	BLE	Zigbee	Z-Wave
Rækkevidde	$\leq$ 100 m	$\leq$ 100 m	$\leq$ 100 m
Data rate	$\leq$ 1 Mbps	$\leq$ 250 kbps	$\leq$ 100 kbps
Kryptering	128-bit AES	128-bit AES	Z-Wave Alliance S2
Enheder	$\leq$ 32.767	$\leq$ 65.536	$\leq$ 232
Hop	$\leq$ 126	Ubegrænset	$\leq$ 4

Z-Wave understøtter kun 232 enheder og 4 hop, hvilket kan være for lidt for en større fremstillingsvirksomhed.

BLE er begrænset til 126 hop, hvilket kan have indflydelse på større fremstillingsvirksomheder med meget lang afstand fra gateway til sidste sensorenhed. Ved så mange enheder, som BLE og Zigbee understøtter, vil det begrænsende element generelt være RAM i gatewayen. Derfor er det ikke et problem at BLE kun understøtter halvt så mange enheder som Zigbee. Den største forskel mellem BLE og Zigbee er at BLE kan sende data 4 gange hurtigere end Zigbee.

Ekstern kommunikation

En fremstillingsvirksomhed har generelt en internetforbindelse, og der vil tit være et tilhørende Wi-Fi. Wi-Fi har hurtig data rate og er relativt billigt at implementere. Med adgang til Wi-Fi og firmaet tillader det IIoT prædiktiv vedligeholdelsessystem at blive tilsluttet gennem deres internetforbindelse vil Wi-Fi være mest egnet.

I tilfælde, hvor der er for stor afstand fra det IIoT prædiktiv vedligeholdelsessystem til en internetforbindelse, og traditionelt Wi-Fi ikke har rækkevidde nok vil Wi-Fi HaLow kunne anvendes.

En Ethernet forbindelse kan også anvendes til at tilslutte det IIoT prædiktiv vedligeholdelsessystem til internettet, hvis virksomheden tillader dette. Idet der kun skal laves én fysisk forbindelse er de ekstra omkostninger i forhold til en trådløs forbindelse marginal.

Er der ikke adgang til en internetforbindelse fra firmaet er cellulær eller LPWAN en mulighed. Ved LPWAN forbindelser som f.eks. Sigfox er der en begrænsning på mængden af data, samt hvor hurtigt dataene kan sendes [11]. 5G og andre cellulære forbindelser er generelt meget dyre, men har ikke samme begrænsninger på datamængder.

Data rate af den eksterne kommunikation skal være mindst lige så hurtig som data rate for den interne kommunikation. Ulicenseret LPWAN's data rate er mindre end data rate for BLE, Zigbee og Z-Wave i følge figur 2.2, derfor sammenlignes maksimal data rate for Ethernet, Wi-Fi, Wi-Fi HaLow, Cellulær og Licenseret LPWAN, se tabel 2.3.

Tabel 2.3: Sammenligning af maksimal data rate for Ethernet, Wi-Fi, Wi-Fi HaLow, Cellulær og Licenseret LPWAN [12] [13] [14] [15] [16] [17] [18].

	Data rate
Ethernet Cat7	$\leq$ 10 Gbps
Wi-Fi 5 GHz	$\leq$ 1.3 Gbps
Wi-Fi 2,4 GHz	$\leq$ 600 Mbps
Wi-Fi HaLow	$\leq$ 80 Mbps
5G	$\leq$ 10 Gbps
4G	$\leq$ 150 Mbps
3G	$\leq$ 7.2 Mbps
LTE-M	$\leq$ 1 Mbps
EC-GSM	$\leq$ 240 kbps
NB-IoT	$\leq$ 200 kbps

Data rate for ekstern kommunikation vil afhænge af bl.a. antal sensorenheder og forventet data rate per sensorenhed. Licenseret LPWAN's data rate er meget lavere end de resterende i tabel 2.3, hvilket i bestemte applikationer kan være et problem.

Matter

Under research af de forskellige trådløse IIoT kommunikationsteknologier blev Matter standarden nævnt flere gange i sammenhængen af at forbedre og simplificerer integreringen af IoT systemer. Matter standarden udvikles af udbyderene af gateways og smart hubs i modsætning til f.eks. Zigbee, der blev udviklet af producenterne af smart enhederne.

Matter blev tidligere kaldet Connected Home over IP (CHIP) og er stadig under udvikling. Standarden er understøttet af bl.a. Samsung, Apple, Amazon og Google. Matter skal gøre integrering af forskellige smartenheder nemmere samtidig med forbedringer af hurtigere responstid, lokal integration og bedre Cyber Security. Matter er baseret på IPv6 og anvender enten IEE 802.3, IEE 802.11 eller IEE 802.15.4 [19].

Matter er primært fokuseret til Smarthjem, hvor der kan anvendes mange forskellige trådløse teknologier på samme tid. Her kan Matter være bindeleddet, der simplificerer opsætningen og brugen af smartprodukter. Standarden forventes færdig i efteråret 2022, hvor der kan forventes færdige produkter i slutningen af 2022 [19].

2.4 Cloud Computing

Ved Cloud Computing købes bl.a. lagerplads, servere, databaser, analyseværktøjer og "Backend Computing" over nettet af en udbyder, hvilket simplificere og fjerner en stor række arbejdsopgaver fra fremstillingsvirksomheden [20].

De største fordele ved Cloud Computing vedrører omkostninger, skalering, ydeevne og sikkerhed.

Udgifter nedsættes ved at der kun betales for de servicer der er nødvendige for opgaven, og der ikke skal investere i infrastrukturen selv, hvilket kan være kostelig [20].

Opskalering af servicer kan gøres elastisk og hurtigt, da der ikke skal investeres i nyt infrastruktur. Dette omhandler opskalering af alt fra lagerplads til båndbredde og geografisk lokation [20].

Udbyderen af Cloud Computing arbejder altid på at forbedre deres ydelser, hvilket betyder at de ofte opgraderer deres infrastruktur. Dette giver løbende forbedringer for fremstillingsvirksomheden uden store omkostninger [20].

Mange udbydere tilbyder en stor række af sikkerhedsforanstaltninger, der gør det simpelt at sikre fremstillingsvirksomhedens data, heriblandt kryptering, overvågning og sikkerhedskopiering af data [20].

De største ulemper ved Cloud Computing vedrører afhængighed, sikkerhed og reaktionstid.

Når der anvendes ydelser fra en Cloud Computing udbyder skabes der en afhængighed af udbyderen. I tilfælde af at udbyderen drastisk ændre eller stopper for ydelser vil dette kunne have store konsekvenser. Derudover kan en applikation, der udvikles under strikse forhold, blive låst og efterfølgende ikke kunne blive åbnet op for.

Når potentiel fortrolig eller følsom data skal sendes uden for firmaets interne netværk rejses der en række ekstra bekymringer vedrørende sikkerhed. Det skal kunne garanteres at dataene ankommer sikkert til udbyderen og at de har de nødvendige sikkerhedsforanstaltninger.

Data transmissions tiden vil generelt være længere når dataene skal sendes over internettet til en Cloud Computing udbyder end ved lokal databehandling. Derfor vil reaktionstiden af det færdige system generelt være større ved anvendelse af Cloud Computing ydelser.

2.4.1 Udbydere

Firebase, Amazon Web Services (AWS) og Azure er tre store udbydere af Cloud Computing ydelser, hvor de tilbyder forskellige ydelsesmodeller. De mest populære ydelsesmodeller er:

Backend as a Service (BaaS) tilbyder database administration, cloud funktionaliteter, bruger godkendelse, hosting og cloud datalagring [21].

Infrastructure as a Service (IaaS) tilbyder cloud-hosted fysiske servere, virtuelle servere, datalagring og netværkshardware [22].

Platform as a Service (PaaS) tilbyder en cloud-hosted platform til at udvikle, kører, vedligeholde og administrer applikationer [22].

Software as a Service (SaaS) tilbyder cloud-hosted klar til brug applikations software [22].

Tabel 2.4 viser en sammenligning af Firebase, AWS og Azure.

Tabel 2.4: Sammenligning af [21].

	Firebase	AWS	Azure
Moderselskab	Google	Amazon	Microsoft
Ydelsesmodeller	BaaS	BaaS, IaaS, PaaS & SaaS	BaaS, IaaS, PaaS & SaaS
Betalingsmodel	Pay-As-You-Go	Pay-As-You-Go	Pay-As-You-Go
Gratis ydelser	10 GB hosting datalagring 5 GB cloud datalagring 10 tests/dag	Gratis altid eller Gratis i 12 måneder	Gratis altid eller Gratis i 12 måneder

Firebase tilbyder kun BaaS og gratis ydelser op til 10 tests/dag, hvilket kan være for begrænsende for udviklingen af et IIoT prædiktivt vedligeholdelsessystem. AWS og Azure tilbyder alle fire ydelsesmodeller og har ydelser, der er gratis altid eller op til 12 måneder.

AWS er generelt nemmere at lære idet deres dokumentation, og guides er bedre formuleret til nye brugere af Cloud Computing ydelser [23].

2.5 Cyber Security

For et IIoT prædiktivt vedligeholdelsessystem vil den interne kommunikation ikke være lige så kritisk som den eksterne kommunikation. For at tilgå den interne kommunikation skal der generelt brydes ind i fremstillingsvirksomheden, hvilket sænker risikoen for utilsigtet adgang til datastrømmen.

Cyber Security er begrænset af det svageste led, da der gennem en enkelt usikker enhed kan skaffes adgang til et sikkert netværk. Det er derfor vigtigt at de systemer, der er tilsluttet virksomhedens netværk ikke er en sikkerhedsrisiko.

BLE anvender 128-bit AES kryptering af datapakken i kommunikation mellem enheder, hvilket ikke kan brydes med "Brute Force Attack" inden for rimelig tid. Derudover anvender BLE "Frequency Hopping", der besværligøre angreb som "Jamming" og "Sniffing" [24].

Den eksterne kommunikation er den mest udsatte idet den kan tilgås eksternt. Derfor skal der bruges flest ressourcer på Cyber Security for at sikre dataene i den eksterne kommunikation.

AWS har en række Cyber Security politikker, der beskytter dataene. Alle enheder skal godkendes før de kan interagere med hinanden. Alt trafik sendes beskyttet over Transport Layer Security (TLS) med f.eks. Message Queuing Telemetry Transport (MQTT), Hypertext Transfer Protocol (HTTP) eller WebSocket protokol, hvor der er 18 mulige måder for kryptering af dataene [25].

IEC 62443

IEC 62443 standarden omhandler Cyber Security, og hvordan det kan implementeres i produkter og virksomheder. Standarden er delt op i ni dele udgivet fra 2007 frem til 2020.

IEC 642443 del 1-1 beskriver syv fundamentale krav til Cyber Security, som er Identification and Authentication Control, Use Control, System Integrity, Data Confidentiality, Restricted Data Flow, Timely Response to Events og Resource Availability [26].

IEC 642443 del 4-2 gennemgår de syv fundamentale krav i forhold til Tekniske sikkerhedskrav til IACS[1]-komponenter. Hvert fundamentale krav har sikkerheds niveauer fra 0 til 4, hvor der ved 0 ikke stilles krav til systemet [26].

[1] International Association of Classification Societies

2.6 Delkonklusion

Dette afsnit er en delkonklusion af sektion 2.1 til 2.5. Herefter beskrives det ideelle IIoT prædiktiv vedligeholdelsessystem ud fra Analysen.

For en produktionsvirksomhed er motivationen for et prædiktivt vedligeholdelsessystem primært økonomisk. Efter en opstarts udgift, vil det betale sig tilbage ved at reducerer mængden og længden af produktionsstop. Samtidig kan der sikres en ensartet standard af produkter, hvilket nedsætter risiko for at miste f.eks. en ISO-9000 certifikation. Et prædiktivt vedligeholdelsessystem har indflydelse på FN's verdensmål 8 idet produktionsprocessen optimeres, og verdensmål 12 idet spild af velfungerende sliddele sænkes.

CbM kan anvendes til prædiktiv vedligeholdelse af elektriske lineær aktuatore, hvor målt data sammenlignes med en kendt model for en fungerende enhed. Det mest anvendte nøgleparameter er 3-akset vibrationsmåling, hvor dataene måles i tids-domænet, og herefter konverteres til frekvens-domænet. Krav til reaktionstiden for systemet afhænger af, hvilke applikationer systemet implementeres i.

Ud fra vibration, strømtræk og temperatur data for en elektriske lineær aktuatore, der er i bevægelse kan det udledes om der er fejl vedrørende kuglelejer, gear, fejljustering, belastningstilstand, excentricitet af rotoren, motorspole, rotor arm, PSU, forøget friktion, forøget belastning og overdreven start/stop. Datamængde til CbM af en elektrisk motor består mest af vibrationsdata, hvor antallet af samples afhænger af motor RPM og ønsket frekvensopløsning.

Kommunikation i et IIoT prædiktiv vedligeholdelsessystem kan opdeles i intern, der omhandler kommunikation mellem sensorenheder og en gateway, og ekstern kommunikation, der omhandler kommunikation mellem gatewayen og en Cloud Computing service.

Trådløse forbindelser er mest hensigtsmæssige at anvende til intern kommunikation idet der under installation og ved vedligeholdelse er mindst tid med produktionsstop. Et Mesh netværk er godt egnet til et IIoT prædiktiv vedligeholdelsessystem da sensorenhederne ikke behøver en direkte forbindelse til gatewayen.

De to mest anvendelige kommunikationsteknologier til intern kommunikation er BLE Mesh og Zigbee, der begge understøtter nok enheder til implementering i en større fremstillingsvirksomhed. Den største forskel mellem dem er at BLE kan sende data 4 gange hurtigere end Zigbee.

Ethernet, Wi-Fi og Cellulær kan alle anvendes til ekstern kommunikation i forhold til data rate. Cellulær kan være meget dyrt, men har god dækning i tilfælde uden adgang til firmaets internetforbindelse. Ethernet og Wi-Fi er simplest og billigst at implementere, hvis firmaet giver lov til at anvende deres internetforbindelse.

Licenseret LPWAN's data rate er meget lavere end Ethernet, Wi-Fi og Cellulær, hvilket i specifikke applikationer kan være problematisk.

Cloud Computing simplificerer arbejdsopgaver og sænker udgifter ved at tilbyde ydelser som f.eks. lagerplads, databaser, analyseværktøjer og "Backend Computing". Sikkerhed og ydeevne øges idet udbyderene løbende arbejder på at forbedre deres ydelser.

Cloud Computing kan skabe afhængighed af en udbyder, hvilket kan være problematisk, hvis udbyderen stopper med at understøtte den anvendte ydelse. Idet dataene skal sendes over internettet til udbyderen rejses der ekstra bekymringer vedrørende sikkerhed. Samtidig vil data transmissions tiden være længere, hvilket vil øge den totale reaktionstid for det færdige system.

AWS, der har Amazon som moderselvskab, tilbyder de fire populære ydelsesmodeller BaaS, IaaS, PaaS og SaaS. AWS har ydelser der er enten gratis for altid eller gratis i op til 12 måneder, og yderlserne er godt dokumenteret med mange veludførte guides. Derudover har AWS en guide til anvendelsen af "Machine Learning" til prædiktiv vedligeholdelse. Dette gør dem samlet set godt egnet til implementering af et IIoT prædiktiv vedligeholdelsessystem.

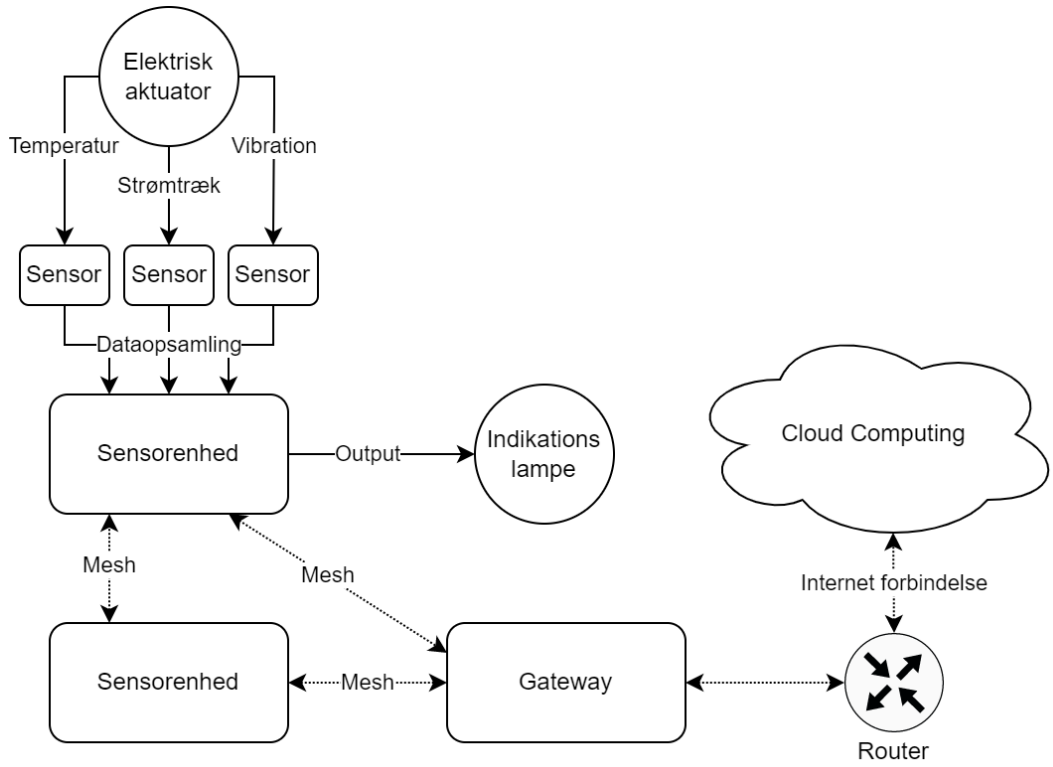
Intern kommunikation er ikke lige så udsat i forhold til Cyber angreb idet der skal brydes fysisk ind i virksomheden. BLE har 128-bit AES kryptering og anvender "Frequency Hopping" til at besværliggøre Cyber angreb.

Ekstern kommunikation er mest udsat for Cyber angreb da den kan tilgås ekstern, hvilket gør det nødvendigt at sikre dataene. AWS anvender kryptering af dataene og alt kommunikation sendes beskyttet over TLS. IEC 62443 standarden beskriver syv fundamentale krav til Cyber Security, og hvordan de kan implementeres i virksomheder og produkter.

2.6.1 Ideelt system

Ud fra Analyse kapitel 2 opstilles der et ideelt IIoT prædiktiv vedligeholdelsessystem. Det ideelle IIoT prædiktiv vedligeholdelsessystem vil bestå af flere sensorenheder, der måler temperatur, strømtræk og vibrationsdata for hver sin elektriske lineær aktuator. Systemets reaktionstid vil være så lavt som muligt for at systemet kan anvendes i flest mulige applikationer.

Sensorenhederne vil være sat op i et Mesh netværk, der kan dække en hele fremstillingsvirksomhedens produktionsareal. Mesh netværket vil have én Gateway, der sender dataene fra sensorenhederene til en Cloud Computing udbyder. Cloud Computing vil anvendes til CbM af dataene fra sensorenhederne, hvorefter status på den lineære aktuator indikeres med en lampe.



Figur 2.4: Ideelt IIoT prædiktiv vedligeholdelsessystem funktionsdiagram.

Figur 2.4 viser et funktionsdiagram for det ideelle IIoT prædiktiv vedligeholdelsessystem.

2.7 Afgrænsning

Dette afsnit gennemgår afgrænsning fra det ideelle IIoT prædiktiv vedligeholdelsessystem beskrevet i afsnit 2.6.1 til et prototype IIoT prædiktiv vedligeholdelsessystem.

Der afgrænses til udvikling af et prototype IIoT prædiktiv vedligeholdelsessystem bestående af én sensorenheder, der måler vibrationsdata for en ubelastet elektriske lineær aktuator når den er i bevægelse.

Afsnit 2.2.2 beskriver at vibrationsdata er den største faktor i datamængden for et IIoT prædiktiv vedligeholdelsessystem. Datamængden, og derved kompleksiteten af transmissionen af dataene, for måling af temperatur, strømtræk og vibrationsdata eller kun vibrationsdata er sammenlignelig.

Til intern kommunikation fravælges Z-Wave grundet den begrænsede mængde enheder og hop. Data rate for BLE og Zigbee sammenlignes i forhold til implementering af et prototype IIoT prædiktiv vedligeholdelsessystem.

Ud fra den udregnede datamængde for vibrationsdata fra ligning 2.5 i afsnit 2.2.2, og Data rate fra tabel 2.2 kan det maksimale antal målinger per sekund udregnes for BLE og Zigbee.

Maksimal antal målinger per sekund for BLE:

$$\left\lfloor \frac{1\,\text{Mbps}}{49{,}15\,\text{kbit}} \right\rfloor = 20\ [\cdot] \tag{2.7}$$

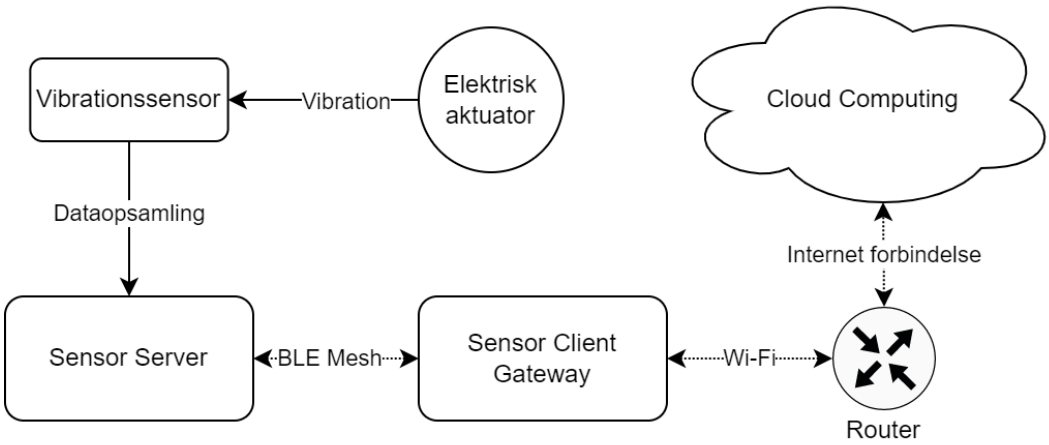
Maksimal antal målinger per sekund for Zigbee:

$$\left\lfloor \frac{250\,\text{kbps}}{49{,}15\,\text{kbit}} \right\rfloor = 5\ [\cdot] \tag{2.8}$$

Da der ikke kendes minimumskravet til antal målinger per sekund vil et BLE Mesh netværk være det sikreste valg til implementering af et prototype IIoT prædiktiv vedligeholdelsessystem. Samtidig er systemet fremtidssikret i forhold til eventuel tilføjelse af ekstra sensorer per sensorenhed.

Sensorenhederne sættes derfor op i et BLE Mesh netværk med en Gateway, der kommunikerer over en Wi-Fi internetforbindelse med AWS Cloud Computing.

Med AWS Cloud Computing skal der laves CbM til et sådan niveau at princippet ved prædiktiv vedligeholdelse vises. Ved målinger uden for specifikation skal der sendes en Fejl Indikations Notifikation.



Figur 2.5: Prototype IIoT prædiktiv vedligeholdelsessystem funktionsdiagram.

Figur 2.5 viser det afgrænsede funktionsdiagram for prototype IIoT prædiktiv vedligeholdelsessystemet ud fra figur 2.4. For at prototype IIoT prædiktiv vedligeholdelsessystemet kan anvendes til flest mulige applikationer ønskes den totale reaktionstid for systemet så lav som muligt.

Der afgrænses fra Cyber Secutity overvejelser da projektets ressourcer er begrænsede, og fokus i projektet er infrastrukturen til et prototype IIoT prædiktiv vedligeholdelsessystem.

2.7.1 Problemformulering

Ud fra afgrænsningen af et prototype IIoT prædiktiv vedligeholdelsessystem er følgende problemformulering opstillet:

Hvorledes kan et prototype IIoT prædiktiv vedligeholdelsessystem udvikles, således at infrastrukturen til og princippet ved prædiktiv vedligeholdelse vises?

Kravspecifikation 3

I følgende kapitel opstilles de vigtigste funktionelle og tekniske krav til prototype IIoT prædiktiv vedligeholdelsessystemet.

Funktionelle krav

De funktionelle krav er udarbejdet i forhold til Afgrænsning afsnit 2.7.

Tabel 3.1: Oversigt over funktionelle krav og dertilhørende referencer.

Krav	Beskrivelse	Reference
F.1	Vibrationsmåling	2.7
F.2	Intern kommunikation BLE Mesh netværk	2.7
F.3	Ekstern kommunikation Wi-Fi internetforbindelse	2.7
F.4	Fejl Indikations Notifikation	2.7

Tekniske krav

Bestemmelsen af tekniske krav T.1 er gennemgået i målejournal A, hvor RPM for den elektriske lineær aktuator først bestemmes. Herefter udregnes 1. til 4. harmoniske frekvenser for den elektriske lineær aktuator, hvilket giver frekvensspændet til krav T.1.

Tabel 3.2: Oversigt over tekniske krav og dertilhørende referencer.

Krav	Beskrivelse	Værdi	Reference
T.1	Vibrationsmålings	13,85 – 55,40 Hz	A

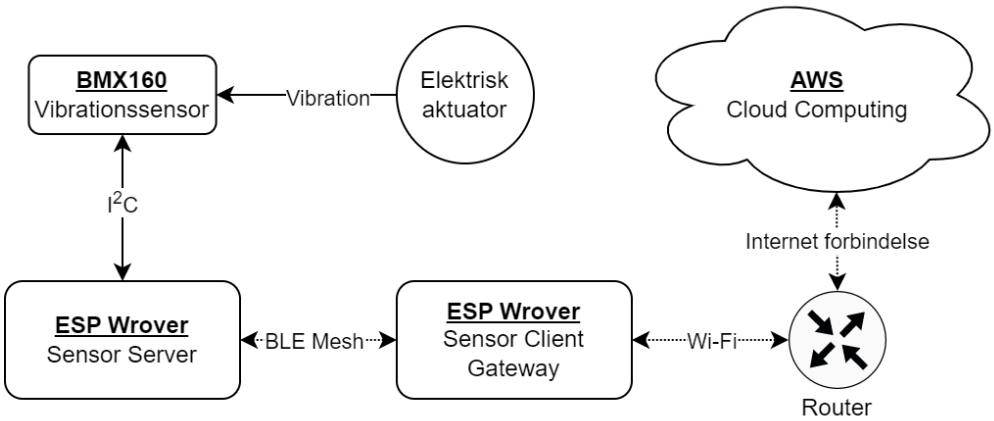
Design 4

Følgende kapitel omhandler design af delsystemerne i prototype IIoT prædiktiv vedligeholdelsessystem bestående af Dataopsamling, BLE Mesh og Cloud Computing.

Dataopsamlings delsystem omhandler måling af vibrationsdata fra den elektriske lineær aktuator. Samplingsfrekvens, antal målinger og datamængde bestemmes i forhold til krav T.1.

BLE Mesh delsystemet omhandler den interne kommunikation i systemet. Information om BLE Mesh netværkstransmission og Sensor Server/Client opsætning og funktionaliteter, samt design af dataopsplitning af vibrationsdata gennemgås.

Cloud Computing delsystemet omhandler den eksterne kommunikation i det prædiktive IIoT vedligeholdelsessystem, samt udførsel af CbM i AWS. Information om Wi-Fi, MQTT, CbM og design af MQTT "Payload" og Fejl Indikations E-mail, samt anvendelse af AWS ydelser gennemgås.



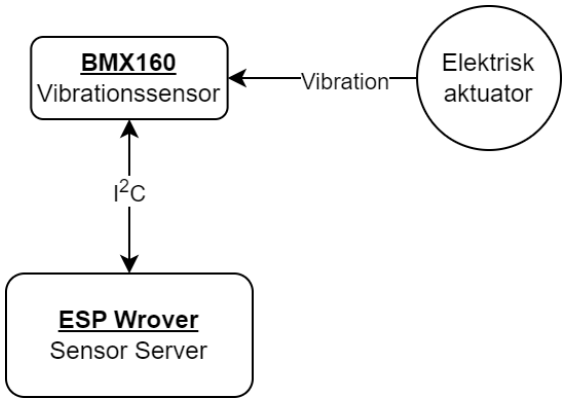
Figur 4.1: Prototype IIoT prædiktiv vedligeholdelsessystem funktionsdiagram.

Figur 4.1 viser funktionsdiagram for prototype IIoT prædiktiv vedligeholdelsessystemet udarbejdet i forhold til figur 2.5 i afsnit 2.7.

Sensor Server og Sensor Client er udarbejdet på hver sit Espressif (ESP) Wrover udviklingsboard. Sensor Serveren måler vibrationsdata gennem et SEN0373 "Breakout Board", der er baseret på en BMX160 9-akset Inertial Measurement Unit (IMU). Valgene af og generelt information om hardware gennemgås i kapitel 5.

4.1 Dataopsamling

Dataopsamlings delsystemet står for måling af vibrationsdata fra den elektriske lineær aktuator, hvorefter dataene senere anvendes til "Condition Monitoring" af Cloud Computing delsystemet. Bestemmelse af samplingsfrekvens, antal målinger og datamængde til vibrationsmåling gennemgås i forhold til krav T.1. Den udregnede totale datamængde anvendes i sektion 4.2.2 til design af BLE Mesh netværks transmission.



Figur 4.2: Dataopsamling funktionsdiagram.

Figur 4.2 viser dataopsamlings delsystemet bestående af Sensor Server (ESP Wrover), der over I^2C kommunikerer med Vibrationssensoren (BMX160).

4.1.1 Vibrationsmåling

Accelerometre i BMX160 anvendes til at måle den g-påvirkning sensoren bliver udsat for over en bestemt periode. Ved at måle g-påvirkningen for X-, Y- og Z-aksen af sensoren dannes der et totalt billede af både påvirkningsgraden og påvirkningsretningen.

Krav T.1 beskriver at IIoT systemet skal kunne måle vibrationer i frekvensspændet $0 - 55{,}40\,\text{Hz}$ (f_n), herudfra kan den mindste samplingsfrekvens (f_s) bestemmes ud fra Shannon Nyquist:

$$f_\text{s} = f_\text{n} \cdot 2 = 55{,}40\,\text{Hz} \cdot 2 = 110{,}80\,\text{Hz} \tag{4.1}$$

Ved at bestemme samplingsfrekvensen til 125 Hz sikres det at der kan måles frekvenser i spændet $0 - 55{,}40$ Hz.

På samme måde som eksemplet i sektion 2.2.2 kan antal målinger og frekvensopløsningen bestemmes. Antal målinger vælges til $2^7 = 128$ målinger.

Målingstiden (D) udregnes ud fra antal målinger (BL) og samplingsfrekvensen (f_s):

$$D = \frac{BL}{f_\mathrm{s}} = \frac{128}{125\,\mathrm{Hz}} = 1{,}024\,\mathrm{s} \tag{4.2}$$

Frekvensopløsningen (df) udregnes:

$$df = \frac{1}{D} = \frac{1}{1{,}024\,\mathrm{s}} \approx 1\,\mathrm{Hz} \tag{4.3}$$

Med en frekvensopløsning på ca. 1 Hz er det muligt at differentiere mellem de forskellige harmoniske frekvenser for vibrationsmåling af den elektriske lineær aktuator.

Accelerometret i BMX160 leverer en 16-bit "Signed" variable for hver akse, der beskriver påvirkningsgraden og påvirkningsretningen. Vibrationer involverer generelt ikke store g-påvirkninger, derfor indstilles accelerometret til at måle i dets mindste spænd på ± 2 g.

Opløsningen af vibrationsmålingen (Opl) udregnes ud fra størrelse af spændet (Spn_Size) og (Var_Size):

$$Opl = \frac{Spn_\mathrm{Size}}{Var_\mathrm{Size}} = \frac{4\,\mathrm{g}}{2^{16}\,\mathrm{bit}} \approx 61\frac{\mu\mathrm{g}}{\mathrm{bit}} \tag{4.4}$$

Datamængden for hver akse ($Data_\mathrm{Akse}$) udregnes ud fra antal målinger (BL) og variable størrelse (Var_Size):

$$Data_\mathrm{Akse} = BL \cdot Var_\mathrm{Size} = 128 \cdot 16\,\mathrm{bit} = 2048\,\mathrm{bit} = 256\,\mathrm{byte} \tag{4.5}$$

Den totale datamængde ($Data_\mathrm{Tot}$) udregnes:

$$Data_\mathrm{Tot} = Data_\mathrm{Akse} \cdot \mathrm{Antal\ Akser} = 256\,\mathrm{byte} \cdot 3 = 768\,\mathrm{byte} \tag{4.6}$$

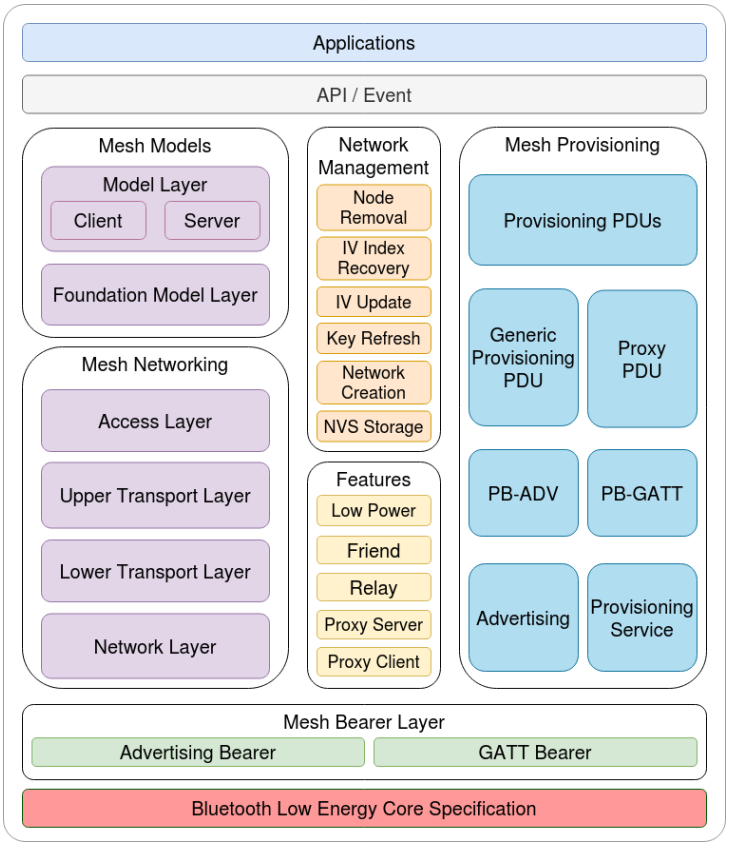
Datamængde for hver akse fra ligning 4.5 og den totale fra ligning 4.6 anvendes i afsnit 4.2.2 til design af BLE Mesh netværks transmission.

4.2 BLE Mesh

BLE Mesh delsystemet omhandler den interne kommunikation i det prædiktive IIoT vedligeholdelsessystem. Baggrundsinformation vedrørende BLE Mesh netværktransmission med opslitning til netværks datapakker, Sensor Server/Client funktionaliteter og opsætning af sensorenheder gennemgås. To tilgange til design af transmission og data-opsplitning af vibrationsdata, samt deres fordele og ulemper gennemgås.

Figur 4.3: BLE Mesh funktionsdiagram.

Figur 4.3 viser BLE Mesh delsystemet bestående af Sensor Server (ESP Wrover), der over BLE Mesh netværk kommunikerer med Sensor Client (ESP Wrover). BLE Mesh er implementeret i ESP Wrover med ESP BLE Mesh stakken, hvor den er implementeret i forhold til Bluetooth standard, og er officielt Bluetooth certificeret.



Figur 4.4: ESP BLE Mesh arkitektur udarbejdet af ESP [27].

Figur 4.4 viser ESP BLE Mesh stakkens arkitektur, der er opdelt i forskellige grupper. Hver gruppe håndterer forskellige typer opgaver, og er opdelt i flere undergrupper. De forskellige gruppers opgaver og funktionaliteter gennemgås i appendiks I.

4.2.1 Baggrundsinformation

Afsnittet gennemgår BLE Mesh Netværkstransmission med lav niveau Network Protocol Data Unit (PDU) og høj niveau Transport PDU. Sensor Server/Client funktionaliteter, opsætning og Sensor Data PDU gennemgås.

Netværktransmission

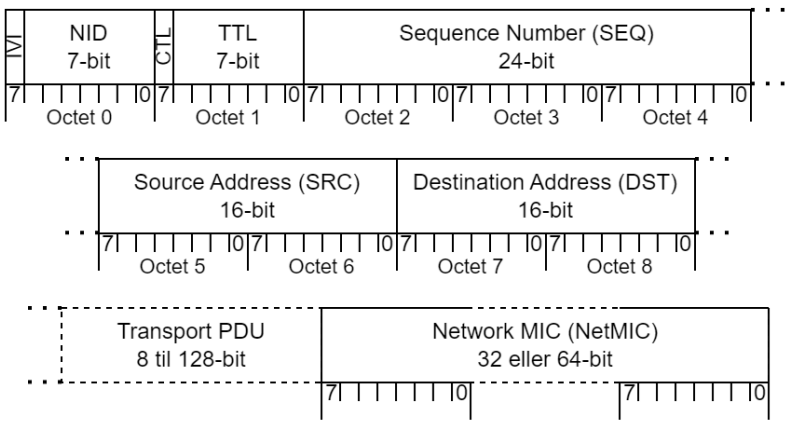
BLE Mesh er baseret på Bluetooth 4.x, der specificerer en underlæggende data rate på 1 Mbps [7]. BLE Mesh datapakken kaldes for Network PDU, og er opdelt i "Octets" hver bestående af 8-bit. Network PDU består af ni dele beskerevet i tabel 4.1.

Tabel 4.1: Network PDU [28].

Navn	Data	Beskrivelse
IVI	1-bit	Anvendes til autentificering og kryptering
NID	7-bit	Identificerer krypterings- og autentificerings-nøgle
CTL	1-bit	Identificerer om datapakke er en "Control" eller "Access Message"
TTL	7-bit	Anvendes til vidersendelse af beskeder for "Relay Nodes"
SEQ	24-bit	Anvendes til at modvirke "Replay Attacks"
SRC	16-bit	Identificerer den "Node", der oprettede denne Network PDU
DST	16-bit	Identificerer destinations "Node" for Network PDU
Transport PDU	8 til 128-bit	Data, der skal transmitteres, 128-bit ved "Access Message" og 96-bit ved "Control Message"
NetMIC	32 eller 64-bit	Autentificerer at "DST" og Transport PDU er uændret, 32-bit ved "Access Message" og 64-bit ved "Control Message"

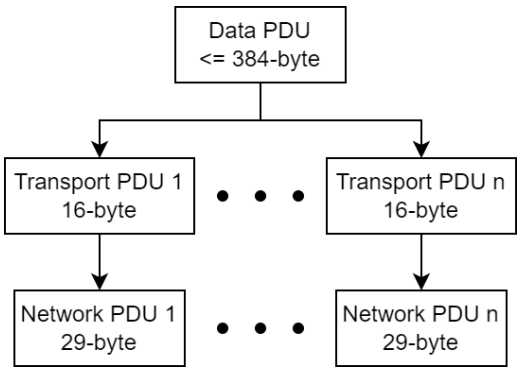
Figur 4.5 illustrerer transmittering af en Network PDU, hvor den samlede størrelse er 29-byte. Transmissionen er opdelt i "Octets", hvor de første syv er kendte, idet størrelserne af delelementerne er faste som beskrevet i tabel 4.1. Transport PDU og NetMIC er af variable størrelse, hvilket gør at det totale antal "Octets" er variabel.

Antallet af retransmissioner indstilles under opsætningen af BLE Mesh "Noden" med et 3-bit parameter. Det er begrænset til maksimalt syv retransmissioner [28].



Figur 4.5: BLE Mesh Network PDU inspireret af [28].

Ved transmissioner af Data PDU'er større end Transport PDU's maksimale størrelse segmenteres datapakken og sendes gennem flere Network PDU'er. Segmentering af datapakken resulterer i en næsten lineær sammenhæng mellem datapakkens størrelse og reaktionstid for systemet ved sending af segmenterede datapakker [7]. Den maksimale størrelse af en Data PDU, der kan sendes med BLE Mesh er 384-byte for "Access Message", og 256-byte for "Control Message" [28].



Figur 4.6: Data PDU opsplitning eksempel.

Figur 4.6 illustrerer opsplitningen af en Data PDU med op til 384-byte til Transport PDU'er af 16-byte, der sendes ved hjælp af 29-byte Network PDU'er.

Sensor Server/Client

En BLE Mesh "Node" opsat som en Sensor Server, anvendes til dataopsamling gennem f.eks. eksterne moduler som vibrationssensoren. Hver Sensor Server kan indeholde op til 38 forskellige sensorenheder, dette er dog begrænset af mængden af data for hver sensorenhed [29].

Sensor Serveren initieres som "Unprovisioned", hvilket betyder at den ikke er tilknyttet et BLE Mesh netværk. Sensor Clienten leder efter "Unprovisioned" "Nodes", og begynder "Provisioning" processen. Efter BLE Mesh netværket er oprettet af Sensor Clienten kan kommunikation mellem Sensor Server og Client begynde.

Sensor Clienten sender en forespørgsel efter Sensor data, og Sensor Serveren modtager dette som et "Event". Sensor Serveren konstruerer en Data PDU, der indeholde sensordata for alle sensorenheder, og sender denne til Sensor Clienten.

Sensorenhederne opsættes individuelt gennem en række parametre. Tabel 4.2 indeholder en oversigt over parametrene til opsætning af en simple sensorenhed.

Tabel 4.2: Sensor opsætningsparametre oversigt.

Parameter	Data	Beskrivelse
Property ID	16-bit	Sensortype og dataformat
Positive Tolerance	12-bit	Grænse for størrelse af en positiv fejlmåling
Negative Tolerance	12-bit	Grænse for størrelse af en negativ fejlmåling
Sampling Function	8-bit	Gennemsnitsudregnings- eller målings-type
Measurement Period	8-bit	Periode for målinger
Update Interval	8-bit	Interval mellem målinger
Sensor Data Format	8-bit	Maksimal sensor data længde og format
Sensor Data Length	8-bit	Sensor data længde
Sensor Raw Value	8-bit	Pointer til buffer med sensor data

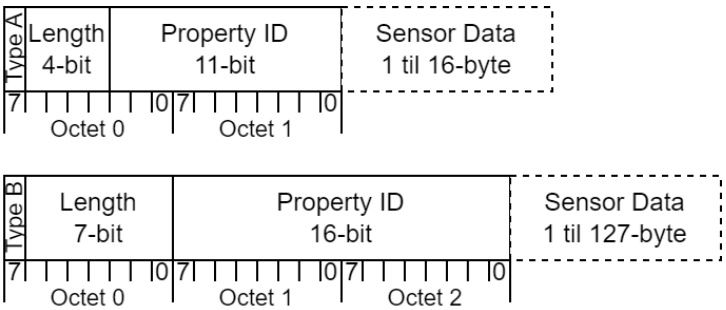
Sensorenhederne skal have hver sin "Property ID", der senere kan anvendes af Sensor Client til at differencere mellem dataene fra de forskellige sensorenheder.

"Positiv Tolerance", "Negative Tolerance", "Sampling Function", "Mesurement Period" og "Update Interval" kan anvendes til at informerer Sensor Client om opsætningen og anvendelsen af sensorenheden.

"Sensor Data Format" beskriver om sensorenheden er type A eller B. Type A anvender 4-bit til at beskrive længden af sensordata i spændet 1 til 16-byte, og 11-bit til "Property ID". Type B anvender 7-bit til at beskrive længden af sensordata i spændet 1 til 127-byte, og 16-bit til "Property ID".

"Sensor Data Length" beskriver længden af sensordata, hvor 0 beskriver en længde på 1-byte, og den maksimale størrelse afhænger af om sensorenheden er type A eller B.

"Sensor Raw Value" er en pointer til den buffer, der indeholde sensordataene, der skal transmitteres.



Figur 4.7: BLE Mesh Sensor Data PDU for type A og B inspireret af [29].

Figur 4.7 illustrerer transmittering af "Sensor Data PDU" for én sensorenhed af type A og B, tilsvarende én Transport PDU.

Ved transmittering af sensordata for flere sensorenheder sammensættes deres "Sensor Data PDU" til én samlet Transport PDU, hvorefter den opsplittes i flere Network PDU'er som vist på figur 4.6.

4.2.2 Design af dataopsplitning

I afsnit 4.1 er det beskrevet at der opsamles vibrationsdata for X-, Y- og Z-aksen. I ligning 4.5 blev den totale datamængde per akse bestemt til 256 byte. Datamængden per akse overskrider den maksimale længde for sensorenheder af type A og B, hvilket gør det nødvendigt at opsplitte vibrationsdataene for de tre akser.

Der er to designtilgange til opsplitningen af vibrationsdataene. Den første tilgang er at opsplitte dataene i flere datatransmissioner, og anvender én sensorenhed per akse. Den anden tilgang er at opsplitte dataene i flere sensorenheder per akse, og sende én datatransmission.

Design 1

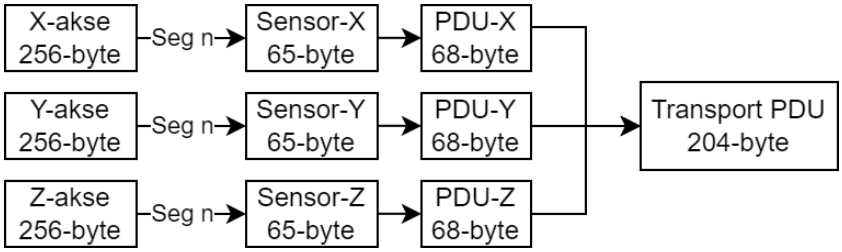
Vibrationsdata for X-, Y- og Z-aksen, oprettes med én sensorenhed per akse. Sensorenhederne opsættes som type B for at reducerer mængden af individuelle datatransmissioner mest muligt.

Vibrationsdatataene opsplittes i lige store dele for at simplificerer datahåndteringen. Da der maksimalt kan sendes 127-byte per sensorenhed er det nødvendigt at opsplitte vibrationsdataene i fire datatransmissioner af 64-byte.

For at sikre at vibrationsdataene rekonstrueres korrekt anvendes den første byte til at beskrive, hvilken del af de opsplittede data, som datapakken tilhører. Dette resulterer i at hver af de tre sensorenheder til vibrationsdata har "Sensor Data Length" på 65-byte.

Fra figur 4.7 vides det, at hver Sensor Data PDU for type B indeholder en "Header" på 3-byte.

Opsplitningsprocessen for én datatransmission er illustreret på figur 4.8, hvor den resulterende Transport PDU er 204-byte.



Figur 4.8: Dataopsplitning Design 1 funktionsdiagram.

Opsplitningsprocessen gentages derefter indtil at alle fire segmenter af vibrationsdataene er transmitteret.

Ved at sende dataene i flere mindre Transport PDU'er kan designet anvendes næsten uanset datamængde, der skal sendes. Idet transmissionstiden er næsten lineær med datamængden vil reaktionstiden af systemet øges i forhold til datamængden.

Grundet at Sensor Clienten initierer hver forespørgsel af sensor data er der risiko for at en af segmenterne ikke når frem. Dette kan resulterer i at Sensor Server og Client kommer ud af synk, så de transmitterede datasæt bliver korrupte. Risikoen for dette er marginalt idet BLE Mesh håndterer retransmission af Network PDU'er.

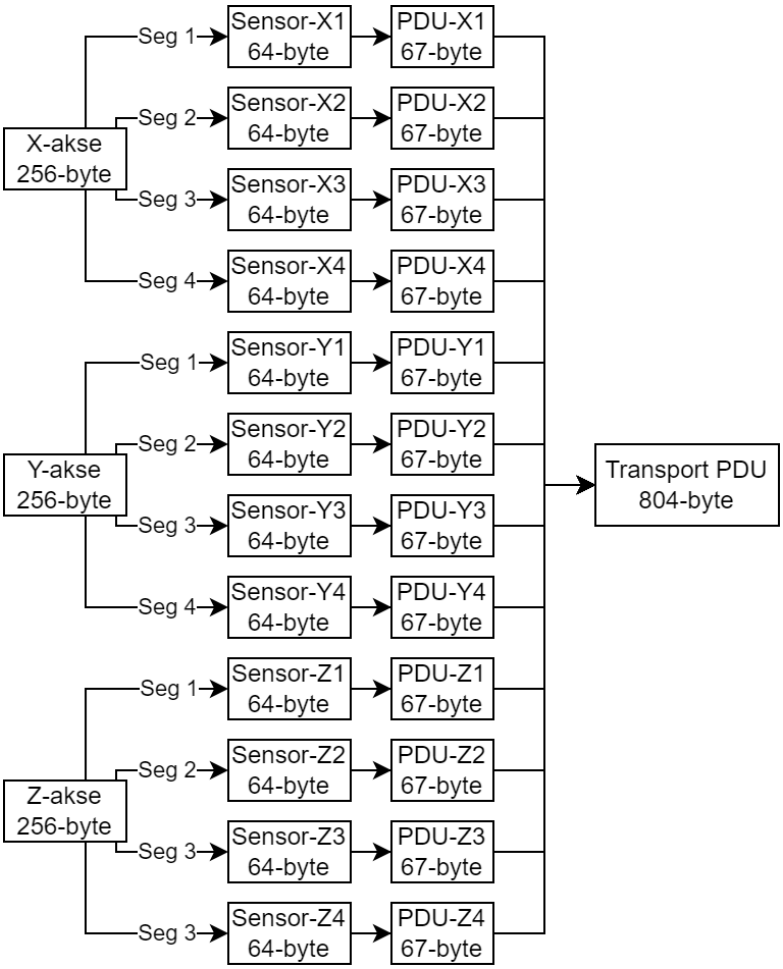
Derudover øges kompleksiteten af datahåndtering på både Sensor Server og Client, idet dataene skal opsplittes og rekombineres korrekt.

Design 2

Vibrationsdata for X-, Y- og Z-aksen, oprettes med fire sensorenheder per akse. Ligesom i Design 1 opsættes sensorenhederne som type B med en 3-byte "Header", og dataene opsplittes i fire dele af 64-byte.

Idet sensorenhederne har hver deres "Property ID" er det allerede indikeret, hvilken del de opsplittede data tilhører.

Opsplitningsprocessen for én datatransmission er illustreret på figur 4.9, hvor den resulterende Transport PDU er 804-byte.



Figur 4.9: Dataopsplitning Design 2 funktionsdiagram.

Ved at transmitterer dataene i en Transport PDU undgås problemer med synkronisering mellem Sensor Server og Client. Samtidig vil datahåndteringen være simplificeret på både Sensor Server og Client idet alt data kan håndteres på en gang.

Da synkroniseringsproblemer undgås og datahåndteringn simplificeres er Design 2 bedre end Design 1. Grundet den maksimale størrelse for en Transport PDU på 384-byte kan Design 2 ikke anvendes når der skal sendes 256-byte vibrationsdata for hver akse.

Prototype IIoT prædiktiv vedligeholdelsessystem implementeres derfor med Design 1 af dataopsplitning.

Sensor Status

Hver gang Sensor Serveren modtager en forespørgsel efter Sensor data fra Sensor Clienten sender den Vibrationsdataene. Da der ikke ønskes at udføres CbM på det samme datasæt flere gange skal det indikeres om det sendte datasæt er nyt.

Til indikation af Sensor Server status oprettes der en sensorenhed af typen B og med længden 1-byte. Sensor Status beskeden er 1-byte stor, hvilket giver mulighed for at indikerer $2^8 = 256$ forskellige statusser.

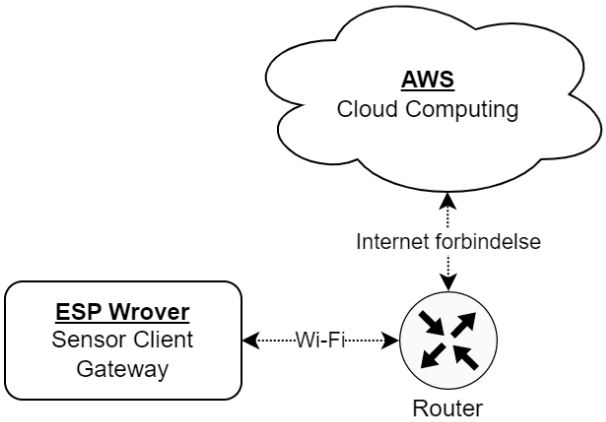
Sensor Status initieres som 0xFF til at indikerer at det Spejlet Datasæt ikke indeholder valide data. Hver gang det Spejlet Datasæt er opdateret med valide vibrationsdata fra en ny måling sættes Sensor Status til 0x00. Efter vibrationsdataene fra det Spejlet Datasæt er sendt over BLE Mesh sættes Sensor Status tilbage til 0xFF.

Den totale størrelse for Data PDU til Sensor Status beskeden inklusiv 3-byte "Header" for type B bliver 4-byte stor. Dette gør at den totale Transport PDU for Sensordata fra Design 1 og Sensor Status beskeden bliver totalt 208-byte. Dette overskrider ikke den maksimale størrelse af Transport PDU på 384-byte.

4.3 Cloud Computing

Cloud Computing delsystemet omhandler den eksterne kommunikation i det prædiktive IIoT vedligeholdelsessystem, samt udførsel af CbM i AWS.

Baggrundsinfo vedrørende Wi-Fi, MQTT og AWS Cloud Computing, samt design af MQTT "Payload" og Fejl Indikations E-mail gennemgås. Anvendelse af FFT til CbM med grafer for vibrationsdata i både tids- og frekvensdomæne, samt sammenligningsværdier til CbM gennemgås. Anvendelse af AWS ydelserne IoT Core, Lambda, Simple Storage Service (S3) og Simple Notification Service (SNS) gennemgås.



Figur 4.10: Cloud Computing funktionsdiagram.

Figur 4.10 viser Cloud Computing delsystemet bestående af Sensor Client (ESP Wrover), der over Wi-Fi sender vibrationsdataene med MQTT til AWS.

4.3.1 Wi-Fi

ESP Wrover har en Wi-Fi stak, der implementerer Wi-Fi 2.4 funktionaliteter herunder fejlhåndtering og genetablering af forbindelse til Wi-Fi, hvis forbindelsen afbrydes. Stakken anvendes på Sensor Clienten til fejlhåndtering af og tilslutning til eksisterende Wi-Fi med internetforbindelse.

For at kunne sende et Tidsstempel for måling af vibrationsdataene opretter Sensor Clienten forbindelse til en Network Time Protocol (NTP) Server, og aflæser det nuværende "Unix" Tidsstempel. Tidsstemplet sendes senere til AWS Cloud Computing, hvor det anvendes til at vise tidspunktet for vibrationsmålingen.

4.3.2 MQTT

MQTT protokollen er meget brugt ved "Machine-To-Machine" kommunikation som smart sensorer eller andre IIoT produkter. Protokollen kan anvendes til tovejs kommunikation af data mellem IIoT produkter og Cloud Computing ydelser [30].

MQTT kræver minimale ressourcer og er meget effektiv, hvilket gør den velegnet til anvendelse på IIoT produkter. Protokollen er skallerbar og kan supporterer forbindelser mellem millioner af enheder. Den indbyggede fejlhåndtering gør det muligt at anvende protokollen i applikationer med en ustabil internetforbindelse [30].

MQTT anvender kryptering af dataudveksling mellem enheder, og kan opsættes til at anvende en af mange forskellige autentificerings protokoller [30].

IIoT enhederne oprettes som MQTT klienter, og Cloud Computing ydelsen agerer som MQTT "Broker", der håndterer autentificering af klienter og koordination af beskeder mellem klienterne [30].

MQTT "Brokeren" filtrerer beskeder baseret på deres tilhørende "Topics". "Brokeren" "Subscriber" til en "Topic", hvorefter alle MQTT beskeder der bliver "Published" med den tilhørende "Topic" læses og sendes videre til datahåndtering [30].

Sensor Client opsættes som en MQTT klient, og AWS IoT Core opsættes som MQTT "Broker". Til autentificering anvendes "Customer Managed Certificates", hvor der genereres et "Root CA Certificate" til identificering af "Brokeren", samt en "Client Private Key" og et "Client Certificate" til identificering af "Clienten".

Design af Payload

MQTT "Payloaden" skal formateres som et JavaScript Object Notation (JSON) objekt, hvor værdi har et tilhørende navn. "Payloaden" består af "Unix" Tidsstempelet samt vibrationsdata for X-, Y- og Z-aksen med målinger 0 til 127. Alle vibrationsdataene tilhørende en akse samles, og indikeres med f.eks. X0 til X127 for X-Aksen.

```
{
  "Time":   1671023642,

  "X0":     -7934,
  ...
  "X127":   -3509,

  "Y0":     2648,
  ...
  "Y127":   891,

  "Z0":     -27820,
  ...
  "Z127":   -28485
}
```

Figur 4.11: Eksempel på struktur af MQTT Payload.

Figur 4.11 viser et eksempel på strukturen af MQTT "Payload", hvor "Unix" Tidsstempelet, samt første og sidste måling for hver akse er vist.

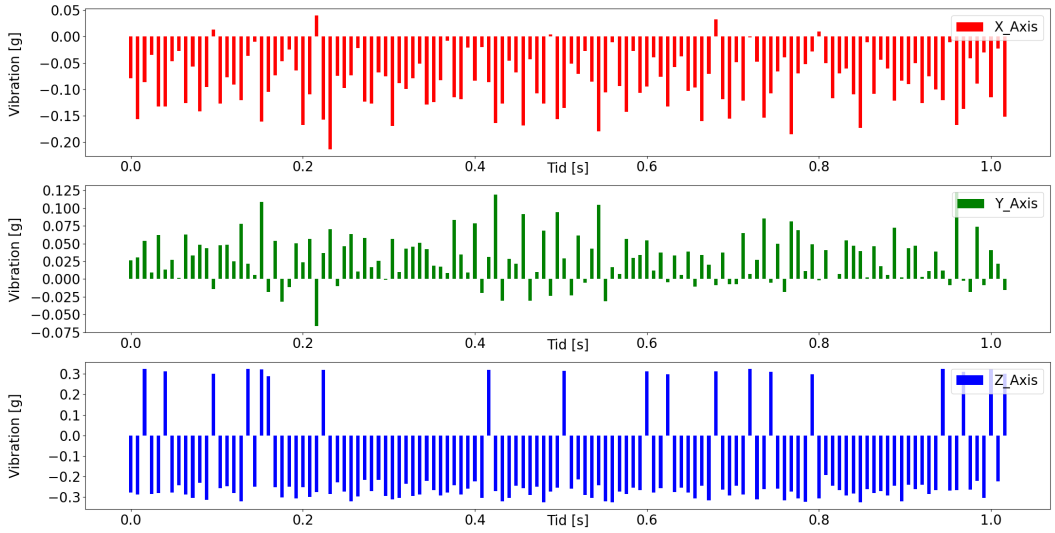
4.3.3 CbM

Som beskrevet i Analyse afsnit 2.2.1 måles vibrationsdataene i tidsdomænet. For at simplificerer databehandling og CbM kan vibrationsdataene transformeres til frekvensdomænet ved f.eks. at udfører FFT på dataene.

Fourier Transform

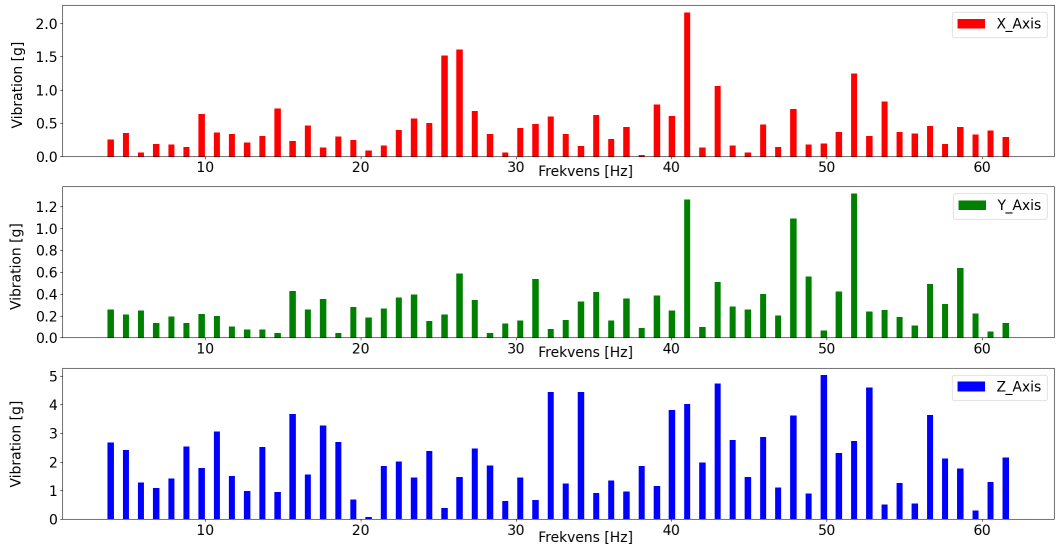
I afsnit 4.1.1 bestemmes samplingsfrekvensen til 125 Hz og antal målinger til 128, hvilket efter FFT giver en frekvensopløsning udregnet i ligning 4.3 til ca. 1 Hz.

Figur 4.12 viser vibrationsdata målt under kørsel med g-påvirkning af den elektriske lineær aktuator for X-, Y- og Z-aksen vist i tidsdomænet. På graferne viser X-aksen tid i sekunder og Y-aksen g-påvirkning i g. Det kan ses at dataene for Z-aksen har større amplitude end X- og Y-aksen, hvilket bl.a. kan komme fra tyngdeaccelerationen, der efter FFT vil fremstå som et DC-offset, hvilket filtreres fra.



Figur 4.12: Vibrationsdata tidsdomæne graf.

Figur 4.13 viser vibrationsdataene efter udførsel af FFT, hvor grafernes X-akse viser frekvens i Hz og Y-aksen g-påvirkning i g. Eventuel DC forskydning på vibrationsdataene fjernes ved at undlade frekvenserne $0 - 4$ Hz.



Figur 4.13: Vibrationsdata frekvenssdomæne graf.

På figur 4.13 kan det for X-aksen ses at der for det gældende vibrations datasæt er "Spikes" omkring de 2. og 3. harmoniske frekvenser. For Y-aksen er der en "Spike" omkring den 2. harmoniske frekvens, dog er denne i ca. samme størrelse som for frekvenserne 48 Hz og 52 Hz. Z-aksen har ikke frekvenser med "Spikes", der skiller sig ud fra de restrerende frekvenser, dog er amplituden som tidligere nævnt generelt større end for X- og Y-aksen.

Tabel A.2 i målejournal A indeholder 1. til 4. harmoniske frekvenser for den elektriske aktuator, der blev bestemt ud fra test.

For at udfører CbM er det nødvendigt at kende de maksimale g-påvirkninger ved de harmoniske frekvenser for en velfungerende elektrisk lineær aktuator.

Idet de harmoniske frekvenser er afhængige af aktuatorens RPM, der kan variere fra aktuator til aktuator, måles et frekvensspænd på ± 2 Hz omkring de harmoniske frekvenser.

Målejournal C gennemgår måling af de maksimale g-påvirkninger ved de harmoniske frekvenser for den elektriske aktuator. Tabel C.6 i målejournalen viser den maksimale målte g-påvirkning for hver akse i frekvensspændende ± 2 Hz omkring de fire harmoniske frekvenser.

Inden dataene fra C.6 kan anvendes til CbM er det nødvendigt at tilføje en ekstra margin til de maksimale målte g-påvirkninger. Den ekstra margin bestemmer følsomheden af CbM sammenligningen, hvor en for høj værdi resulterer i at der reageres for sent på fejl, og en for lav værdi resulterer i at der sendes unødvendigt tidlige fejlbeskeder.

Som beskrevet i Analyse afsnit 2.2.1 vil fejl vedrørende bl.a. kuglelejer og gear for den elektriske lineær aktuator fremstå som 1. til 4. harmoniske frekvenser af aktuatorens RPM.

Ved slid på et kugleleje eller gear vil dette fremstå ved at amplituden ved en af de harmoniske frekvensen stiger. Desto mere slidt kuglelejet eller gearet bliver desto større vil amplituden blive.

Ved test vil det kunne bestemmes, hvilke af de harmoniske frekvenser, samt deres amplitude, der måles når kugleleje eller gear er slidt. Den ekstra margin vil derved kunne bestemmes, så systemet reagerer når amplituden overskrides ved de harmoniske frekvenser.

Ved at sammenligne g-påvirkninger ved de harmoniske frekvenser for alle nye vibrationsmålinger med dataene fra C.6 inklusiv den ekstra margin, kan der bestemmes om den elektriske lineær aktuator har begyndende behov for service.

Fejl Indikation

Når der ved hjælp af CbM bliver registreret fejl på vibrationsdataene skal denne indikeres til en serviceperson, der efterfølgende kan udbedre problematikken med den elektriske lineær aktuator.

Ved opståede fejl oprettes der en E-mail indeholdende navn på enhed med fejl, Tidsstempel for vibrationsmålingen, samt de kombinationer af akser og harmoniske frekvenser, der blev observeret uden for specifikation.

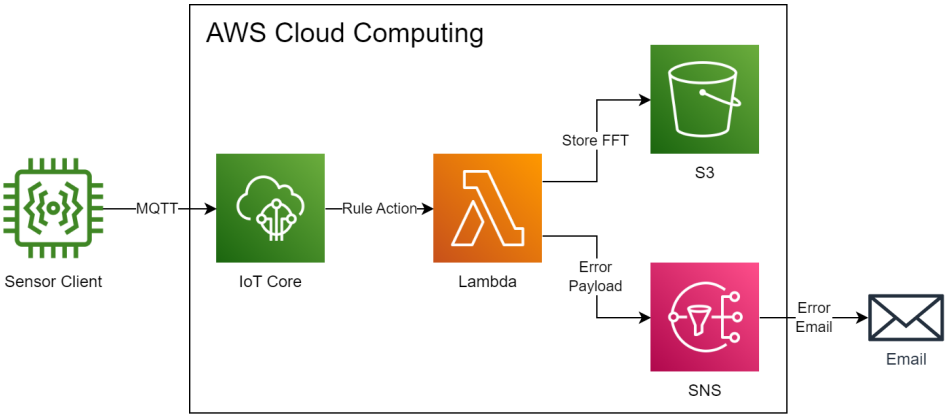
IIoT Predictive Maintenance Unit 01
Timestamp:
12/14/2022 14:57:27
Errors:
X-Axis 3. Harmonic Error
Z-Axis 3. Harmonic Error

Figur 4.14: Eksempel på struktur af Fejl Indikations E-mail.

Figur 4.14 viser et eksempel på strukturen af Fejl Indikations E-mail.

4.3.4 AWS Cloud Computing

AWS tilbyder en række forskellige ydelser som beskrevet i Analyse afsnit 2.4. Cloud Computing delsysstemet skal kunne modtage vibrationsdataene sendt med MQTT, hvorefter der skal udføres FFT og CbM på dataene. Vibrationsdataene før og efter FFT skal gemmes, så de er tilgængelige til eventuel efterbehandling. Ved målinger uden for specifikation skal der oprettes en E-mail med en fejlbesked, der kan sendes til en serviceperson.



Figur 4.15: Funktionsdiagram for Sensor Client og AWS Cloud Computing med ikoner fra [31].

Figur 4.15 viser et funktionsdiagram for AWS Cloud Computing bestående af AWS ydelserne IoT Core, Lambda, S3 og SNS.

IoT Core ydelsen er en "Message Broker" beregnet til implementering af IoT systemer. Ydelsen anvendes som MQTT "Broker", og til oprettelse af de nødvendige autentificerings nøgler og certifikater. Samtidig oprettes der en "Rule", der "Subscriber" til den "Topic", hvor MQTT beskeden bliver "Published", og sender beskeden videre til Lambda ydelsen.

Lambda ydelsen er en "Event-driven" BaaS, der kører et kodestykke hver gang der opstår en pre-defineret "Event". Ydelsen anvendes til at restrukturere vibrationsdataene, og udfører FFT, samt CbM. Ydelsen sender vibrationsdataene før og efter FFT til S3 ydelsen. Ved målinger uden for specifikation oprettes en E-mail "Payload", der sendes til SNS ydelsen.

S3 ydelsen er en "Object Storage", der kan opbevare filer. Ydelsen gemmer de modtagede vibrationsdata fra Lambda i to csv filer, der efterfølgende kan tilgås.

SNS ydelsen er en "Notification Service", der kan sende notifikationer over flere forskellige protokoller. Ydelsen sender den modtagede E-mail "Payload" fra Lambda til en specificeret E-mail adresse.

Implementering 5

Følgende kapitel omhandler implementering af delsystemerne i prototype IIoT prædiktiv vedligeholdelsessystem bestående af Dataopsamling, BLE Mesh og Cloud Computing. Dette gøres ud fra de designvalg, der er blevet gennemgået i Design kapitel 4.

Hardware og udviklingsmiljø til en fysiske prototype bestående af Sensor Server, Vibrationssensoren og Sensor Client gennemgås. Beskrivelse af komponenter og udviklingsmiljø, samt valg af komponenter og udfordringer ved udviklingsmiljø gennemgås.

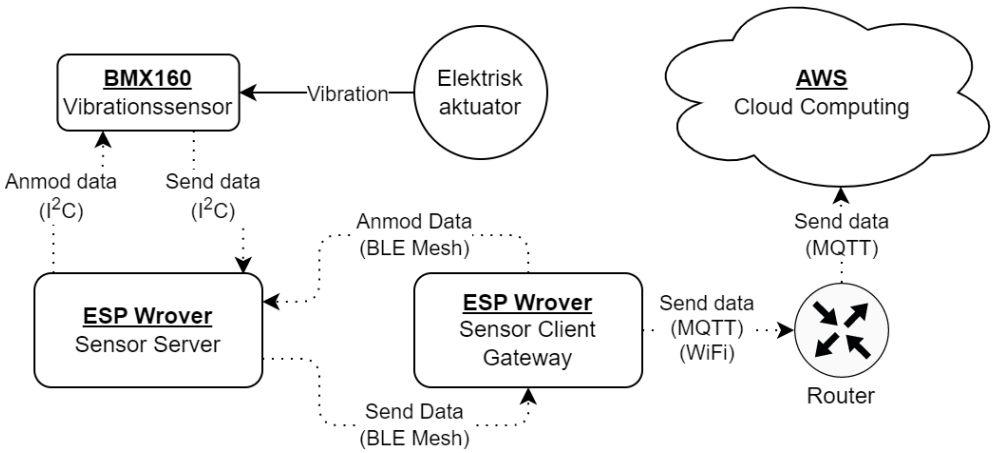
Implementering af Datahåndtering, Dataopsamling, BLE Mesh og Real-Time Operating System (RTOS) for Sensor Serveren ud fra design af Dataopsamling beskrevet i afsnit 4.1, og BLE Mesh med Design 1 til opsplitning af datatransmission beskrevet i afsnit 4.2 gennemgås.

Implementering af Datahåndtering, BLE Mesh, Wi-Fi, AWS MQTT og RTOS for Sensor Clienten ud fra BLE Mesh med Design 1 til opsplitning af datatransmission beskrevet i afsnit 4.2, og Wi-Fi samt design af MQTT "Payload" i afsnit 4.3 gennemgås.

Implementering af IoT Core, S3, SNS og Lambda gemmengås for Cloud Computing ud fra design af AWS Cloud Computing med design af CbM og Fejl Indikations E-mail beskrevet i afsnit 4.3 gennemgås.

Kode til Sensor Server, Sensor Client og AWS Lambda funktion er uploaded til GitHub, og kan findes på følgende link:

https://github.com/Fbang18aau/IIoT_PredictiveMaintenance

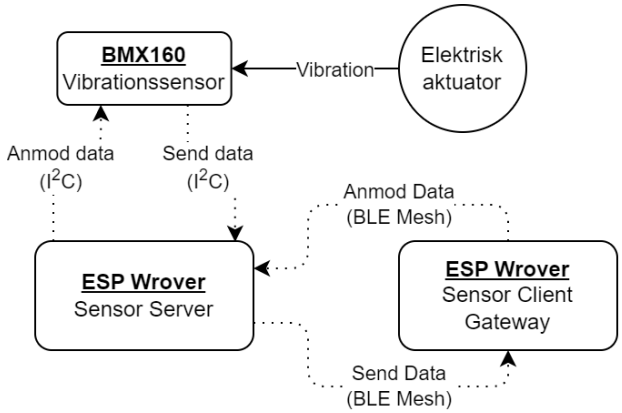


Figur 5.1: Prototype IIoT prædiktiv vedligeholdelsessystem kommunikationsdiagram.

Figur 5.1 viser et kommunikationsdiagram for prototype IIoT prædiktiv vedligeholdelsessystem, hvor typen og retningen af kommunikationen mellem de forskellige delsystemer er anvist.

5.1 Fysisk prototype

Afsnittet gennemgår hardware og udviklingsmiljø anvendt til udvikling af den fysiske del af prototype IIoT prædiktiv vedligeholdelsessystem.



Figur 5.2: Fysiske prototype kommunikationsdiagram.

Figur 5.2 viser et kommunikationsdiagram for den fysiske del af prototype IIoT prædiktiv vedligeholdelsessystemet bestående af Sensor Server (ESP Wrover), Vibrationssensoren (BMX160) og Sensor Client (ESP Wrover).

5.1.1 Hardware

Hardware til den fysiske del af prototype IIoT prædiktiv vedligeholdelsessystemet består af Sensor Server, Sensor Client og Vibrationssensor.

Sensor Server/Client

ESP produkter er meget anvendte inden for IoT, hvor de anvendes i bl.a. Smart Relæer og Smart Sensorer. ESP produkter supporterer flere forskellige trådløsekommunikationsformer heriblandt Bluetooth og Wi-Fi. ESP har meget dybdegående dokumentation, der bl.a. beskriver opsætning og anvendelse af deres færdige stakke.

ESP samarbejder med en lang række kendte firmaer om bl.a. Cloud Computing som Amazon AWS, Microsoft Azure og Google Cloud Platform. Herved kan det forventes at ESP produkter er oppe til samarbejdspartners standard [32].

ESP har integrering af en RTOS stak kalder FreeRTOS, der er open source og beregnet til implementering på mikroprocessorer og System on a Chip (SoC). FreeRTOS anvendes i mange IIoT produkter, hvor stakken er integreret i produkter fra ESP, Texas Instruments, Microchip og andre store firmaer.

Til implementering af Sensor Server og Client vælges et ESP Wrover udviklingsboard til hver, da dette er en anerkendt platform inden for IoT produkter, der er dybdegående dokumentation og at ESP samarbejder med en række kendte store firmaer. Udviklingsboard anvender en dual-core 240 MHz ESP32 Wrover SoC. Det understøtter BLE og Wi-Fi 2.4 trådløskommunikation, hvor ESP tilbyder færdige stakke, der anvender FreeRTOS til implementering heraf.

ESP Wrover udviklingsboardet har indbygget Joint Test Action Group (JTAG), der kan anvendes til hardware debugging. Hardware debugging anvendes ofte under udvikling af software til Embedded systemer, hvor det bl.a. giver mulighed for at “Steppe” igennem koden linje for linje.

Vibrationssensor

Bosch BMX160 er en 9-akset IMU, der anvendes i bl.a. smartphones, smartwatches og kamerare. BMX160 har et lavt strømtræk og lille fysisk størrelse, hvilket gør den velegnet til anvendelse i IIoT produkter. Bosch indeholder et 3-akset accelerometer, et 3-akset gyroskop og en 3-akset geomagnetisk sensor. Accelerometret i BMX160 har en relativ høj opløsning på 16-bit til beskrivelse af g-påvirkning i spændene ±2 g, ±4 g, ±8 g eller ±16 g. BMX160 kan kommunikeres med over I^2C eller SPI, og har dybdegående dokumentation vedrørende opsætning og anvendelse.

Til implementering af vibrationssensor anvendes accelerometret i et SEN0373 "Breakout Board", der er baseret på BMX160 IMU, da den er velegnet til IIoT produkter, har en lille fysisk størrelse, accelerometret har en relativ høj opløsning og den giver mulighed for at implementerer gyroskop eller geomagnetisk sensor efterfølgende.

5.1.2 Udviklingsmiljø

Til softwareudvikling af den fysiske del af prototype IIoT prædiktiv vedligeholdelsessystemet anvendes udviklingsmiljøet Visual Studio Code (VC Code) udviklet af Microsoft. VC Code er velegnet til udvikling af Embedded software, hvor hardware debugging kan anvendes til fejlfinding og test.

ESP officielle IoT Development Framework (IDF) er beregnet til softwareudvikling på bl.a. ESP32 Wrover SoC. ESP IDF indeholder en stor række software komponenter som f.eks. FreeRTOS, BLE Mesh og Wi-Fi stakkene. For at anvende VC Code til softwareudvikling til ESP32 Wrover SoC skal ESP IDF installeres som en "Extension" [33].

Under opsætning af hardware debugging i VC Code opstod en række problemer, der ledte til ustabiliteter. Dokumentationen til opsætningsprocessen af hardware debugging er mangelfuld, idet opsætningen skal laves specifikt til det anvendte udviklingsboard.

Når hardware debugging initialiseres i VC Code tilslutter udviklingsmiljøet til ESP32 Wrover SoC'en gennem JTAG interface. Som standard er SoC'en opsat til "Startup From Reset", hvilket betyder at koden begynder at eksekverer når den genstartes eller tilsluttes forsyning. Denne funktionalitet er nødvendig når produktet skal udgives, men det gør at hardware debugging interfacet ikke når at tilslutte til SoC'en før koden eksekveres.

For at løse problematikken tilføjes et manuelt stop i koden, hvilket gør at koden kun kan eksekveres når der tilsluttes med JTAG og anvendes hardware debugging. Dette er tilsvarende at "Startup From Reset" slåes fra, hvilket sørger for at hardware debugging interfacet altid når at tilslutte til SoC'en.

5.2 Sensor Server

Sensor Serverens to hovedfunktionaliteter består af Dataopsamling og transmission af sensordata over BLE Mesh netværk. Håndteringen af dataene så der undgås korruption af datasættet, implementering af dataopsamlings delsystemet, implementering af BLE Mesh kommunikation og FreeRTOS "Tasks" tilhørende Sensor Serveren gennemgås.

5.2.1 Datahåndtering

Dataopsamlingen af vibrationsdata for X-, Y- og Z-aksen tager ca. 1 sekund, hvor dataene løbende gemmes i et samlet datasæt. Det er vigtigt at vibrationsdatasættet sendes som et komplet datasæt, hvor det er sikret at alle dataene kommer fra samme vibrationsdataopsamling.

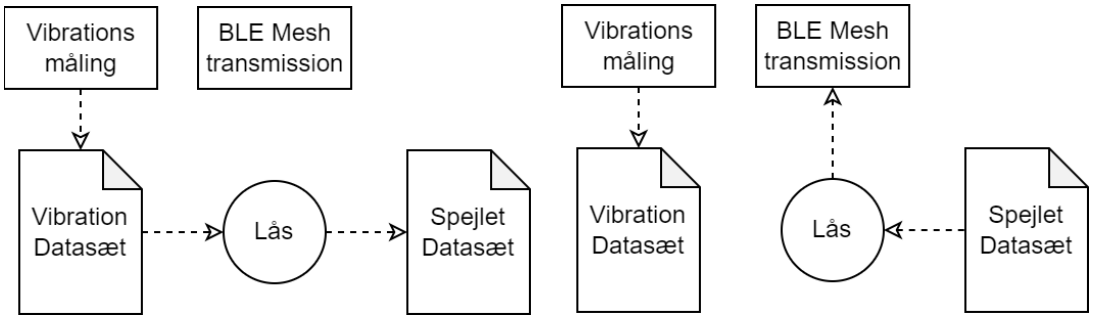
Sensor Serveren anvender FreeRTOS, og er opdelt i "Tasks", der beskrives i afsnit 5.2.4. "Tasks" kører generelt asynkront, hvilket betyder at der kan opstå situationer, hvor transmission af datasættet påbegyndes under dataopsamling. Dette vil resulterer i at der transmitteres data fra to forskellige dataopsamlinger, hvilket er ensbetydende at datasættet er korrupt og ubrugeligt.

Implementering

For at undgå korruption kan der anvendes et låst Spejlet Datasæt, der kun kan tilgås ved brug af en nøgle.

Når dataopsamlingen af vibrationsdata er færdig tages nøglen, hvis den er ledig ellers ventes der på at nøglen bliver ledig. Derefter opdateres det Spejlet Datasæt, og nøglen frigives.

Før transmission af det Spejlet Datasæt over BLE Mesh tages nøglen, hvis den er ledig ellers ventes der på at nøglen bliver ledig. Efter hele datasættet er transmitteret frigives nøglen.



(a) Dataopsamling har nøglen. ***(b)*** BLE Mesh har nøglen.

Figur 5.3: Anvendelse af det låste Spejlet Datasæt.

Figur 5.3 illustrerer anvendelsen af det låste Spejlet Datasæt, hvor de stiplede pile indikerer dataoverførsel. På figur 5.3a har dataopsamling nøglen, hvilket gør det muligt at opdaterer det Spejlet Datasæt med data fra Vibrations Datasættet. På figur 5.3b har BLE Mesh nøglen, hvilket gør det muligt sende det Spejlet Datasæt over BLE Mesh.

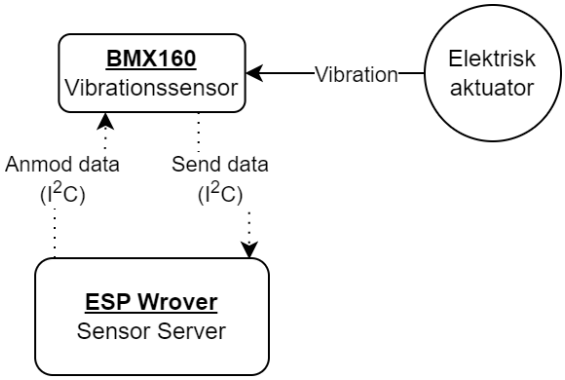
Der sættes en maksimal tid, der ventes på at nøglen bliver ledig, før der meldes at nøglen ikke kunne tages. Dette gøres for at sikre at systemet ikke venter for evigt i situationer med eventuelle fejl i frigivelsen af nøglen.

Maks. ventetid for BLE Mesh sættes til 100 ms ud fra opdateringstiden af det Spejlet Datasæt inklusiv en ekstra margin. Maks. ventetid for opdatering af det Spejlet Datasæt sættes til 6 s udfra transmissionstiden af hele datasættet over BLE Mesh inklusiv en ekstra margin.

Til implementering af det låste Spejlet Datasæt anvendes FreeRTOS stakken til at oprette en “Mutex” som lås. Denne kan derefter tages og frigives gennem funktioner beskrevet i FreeRTOS stakken.

5.2.2 Dataopsamling

Afsnittet gennemgår implementering af Dataopsamlings delsystemet, hvor vibrationsopløsning, reaktionstid for Vibrationssensor og Vibrationsmålings rutinen gennemgås.



Figur 5.4: Dataopsamling kommunikationsdiagram.

Figur 5.4 viser et kommunikationsdiagram for Dataopsamlings delsystemet bestående af Sensor Server (ESP Wrover), og Vibrationssensoren (BMX160). Sensor Server anmoder Vibrationssensoren om data over I^2C, hvorefter Vibrationssensoren sender dataene.

Der oprettes en driver til Vibrationssensoren, der indeholder de nødvendige funktioner for initialisering og anvendelse. Accelerometret i BMX160 initialiseres til at måle i det mindste spænd på ± 2g. Ved anmodning om vibrationsdata fra Vibrationssensoren leveres en 16-bit variable for hver akse.

I afsnit 4.1 blev samplingsfrekvensen bestemt til 125 Hz, og antal målinger bestemt til 128 målinger. Der oprettes en rutine til at sample dataene fra Vibrationssensoren ud fra samplingsfrekvensen og antal målinger.

Samplingsfrekvensen (f_s) omregnes til den totale tid (V_{Tot}), der skal ventes mellem anmodningerne om vibrationsdata:

$$V_{\text{Tot}} = \frac{1}{f_s} = \frac{1}{125\,\text{Hz}} = 8\,\text{ms} \tag{5.1}$$

Reaktionstiden af Vibrationssensoren skal være inkluderet i de 8 ms for at sikre at samplingsfrekvensen er 125 Hz. Reaktionstiden findes ved test og gennemgås i målejournal B, hvor reaktionstid for Vibrationssensoren måles og den gennemsnitlige reaktionstid udregnes til 352 μs.

Den resulterende ventetid mellem målinger (V_{Res}) udregnes ud fra den gennemsnitslige reaktionstid (R_{Gns}) og den totale ventetid (V_{Tot}):

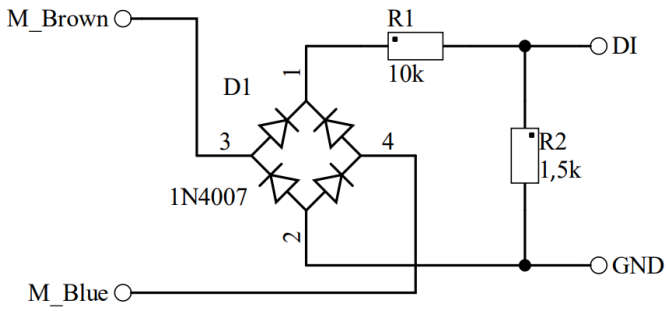
$$V_{\text{Res}} = V_{\text{Tot}} - R_{\text{Gns}} = 8\,\text{ms} - 352\,\mu\text{s} = 7{,}648\,\text{ms} \tag{5.2}$$

Den udregnede resulterende ventetid anvendes som ventetid mellem målinger i Vibrationsmålings rutinen.

Den elektriske lineær aktuator har to forsyningsledninger (brun og blå), hvor bevægelsesretningen ændres ved at ændre polariseringen af forsyningen. Ved at tilslutte den brune ledning til 25 VDC og den blå ledning til GND bevæger aktuatoren sig frem, ved modsat polarisering bevæger den sig tilbage.

Implementering

For kun at måle vibrationsdata når aktuatoren er i bevægelse kan forsyningsspændingen derfor måles for at bestemmes om den er i bevægelse. Et 3,3 VDC digitalt input på ESP Wrover anvendes til at måle om aktuatoren forsynes eller ej.

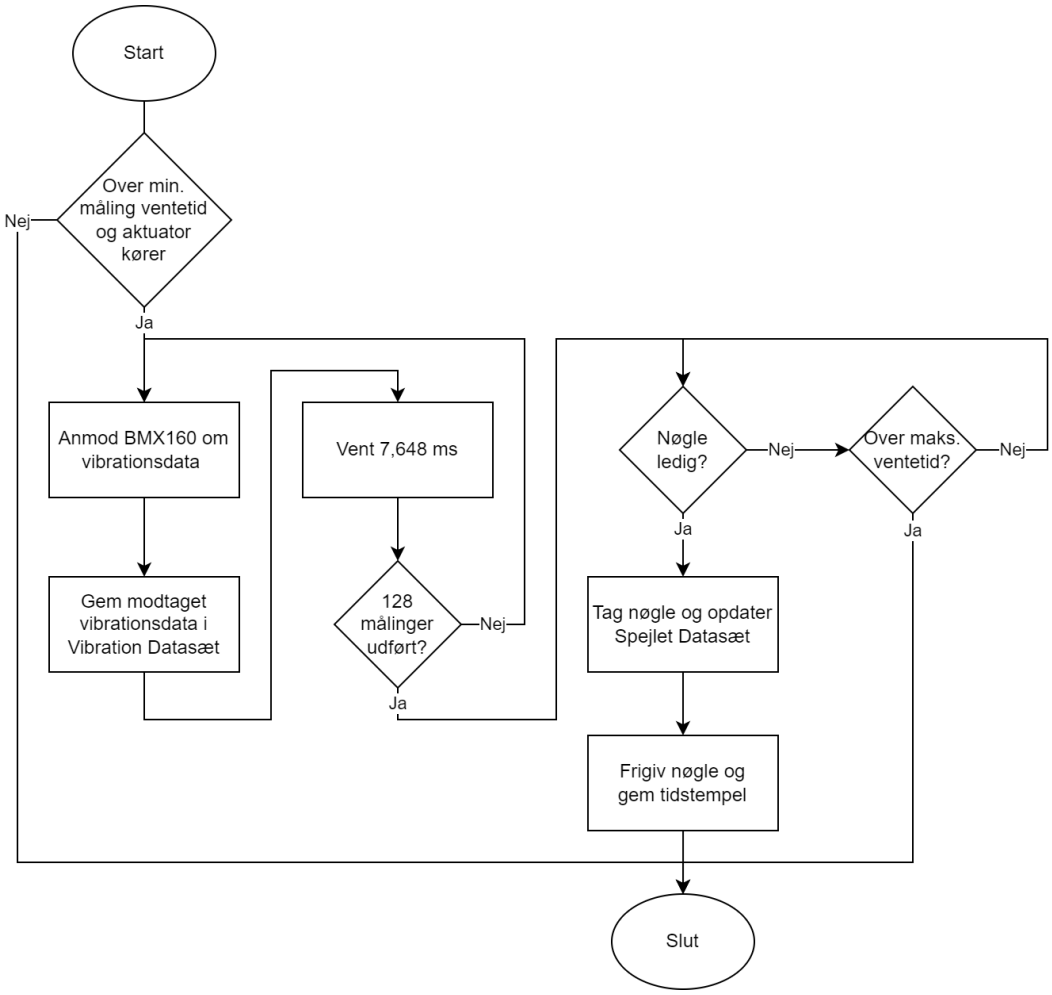


Figur 5.5: Målingskredsløb til beskyttelse af digital input.

For at sikre det digitale input anvendes en spændingsdeler til at sænke spændingen fra 25VDC til 3,3VDC. Derudover anvendes fire dioder som en ensretterbro til så det digitale input modtager en positiv spænding uanset polarisering af aktuatorforsyningen. Figur 5.5 viser kredsløbet til beskyttelse af det digitale input.

Der sættes en minimum ventetid på 20 s mellem måling af vibrationsdata, hvilket anvendes til at sikre at der ikke fortages unødvendigt mange målinger når aktuatoren er i bevægelse.

Der oprettes en Vibrationsmålings rutine til at opsamle vibrationsdata fra BMX160, og efterfølgende gemme dem i et Spejlet Datasæt. Figur 5.6 viser flowchart for Vibrationsmålings rutinen, hvor der udføres 128 målinger, og ventetiden mellem målinger er 7,648 ms.



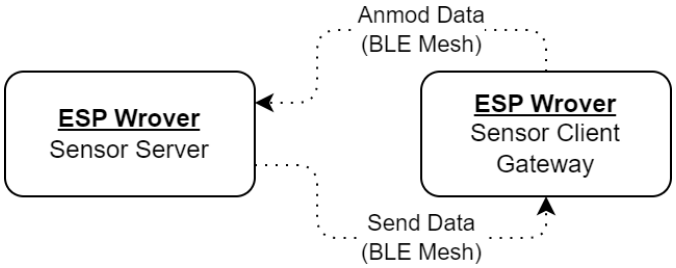
Figur 5.6: Flowchart for vibrationsmålings rutine.

Hver gang Vibrationsmålings rutinen køres undersøges det først om der er gået over 20 s mellem målinger samt at aktuatoren kører. Herefter anmodes vibrationssensoren (BMX160) om vibrationsdata, hvorefter de modtagede data gemmes i et Vibrations Datasæt. Der ventes i 7,648 ms, og datasamplingsprocessen gentages.

Efter 128 målinger tages nøglen til det Spejlet Datasæt, hvis den er ledig, ellers ventes i op til 6s før forsøget fejler (bestemt i afsnit 5.2.1). Efter nøglen er taget opdateres det Spejlet Datasæt med dataene fra Vibrations Datasættet, nøglen frigives og Vibrationsmålings rutinen afsluttes.

5.2.3 BLE Mesh

Afsnittet gennemgår implementering af BLE Mesh kommunikation på Sensor Serveren, hvor opsætningen af sensorenheder, initialisering af Sensor Server og BLE Mesh Data-transmission rutine gennemgås.



Figur 5.7: BLE Mesh kommunikationsdiagram.

Figur 5.7 viser et kommunikationsdiagram for BLE Mesh delsystemet bestående af Sensor Server (ESP Wrover) og Sensor Client (ESP Wrover). Sensor Client anmoder Sensor Server om et segment af Sensordataene over BLE Mesh, hvorefter Sensor Server sender datasegmentet. Dette gentages indtil alle segmenter er sendt.

Opsætningen af sensorenhederne gøres i forhold til Design 1 beskrevet i afsnit 4.2, hvor der oprettes én sensorenhed per akse til vibrationsdataene og én sensorenhed til indikering af Sensor Status.

De fire sensorenheder opsættes ens bortset fra deres "Sensor Data Length", der er af forskellige størrelser, og "Property ID", der skal være unik for hver sensorenhed. "Positiv Tolerance", "Negative Tolerance", "Sampling Function", "Mesurement Period" og "Update Interval" opsættes som "Unspecified", og anvendes ikke.

Sensorenhederne til vibrationsdata oprettes som bestemt i Design 1 beskrevet i afsnit 4.2.2, hvor "Sensor Data Format" sættes til Type B og "Sensor Data Length" sættes til 65-byte. "Property ID" opsættes for X-Aksen til 0x90, Y-Aksen til 0x91 og Z-Aksen til 0x92.

Sensorenheden til Sensor Status oprettes som beskrevet i afsnit 4.2.2, hvor "Sensor Data Format" sættes til Type B, "Sensor Data Length" sættes til 1-byte og "Property ID" sættes til 0x93.

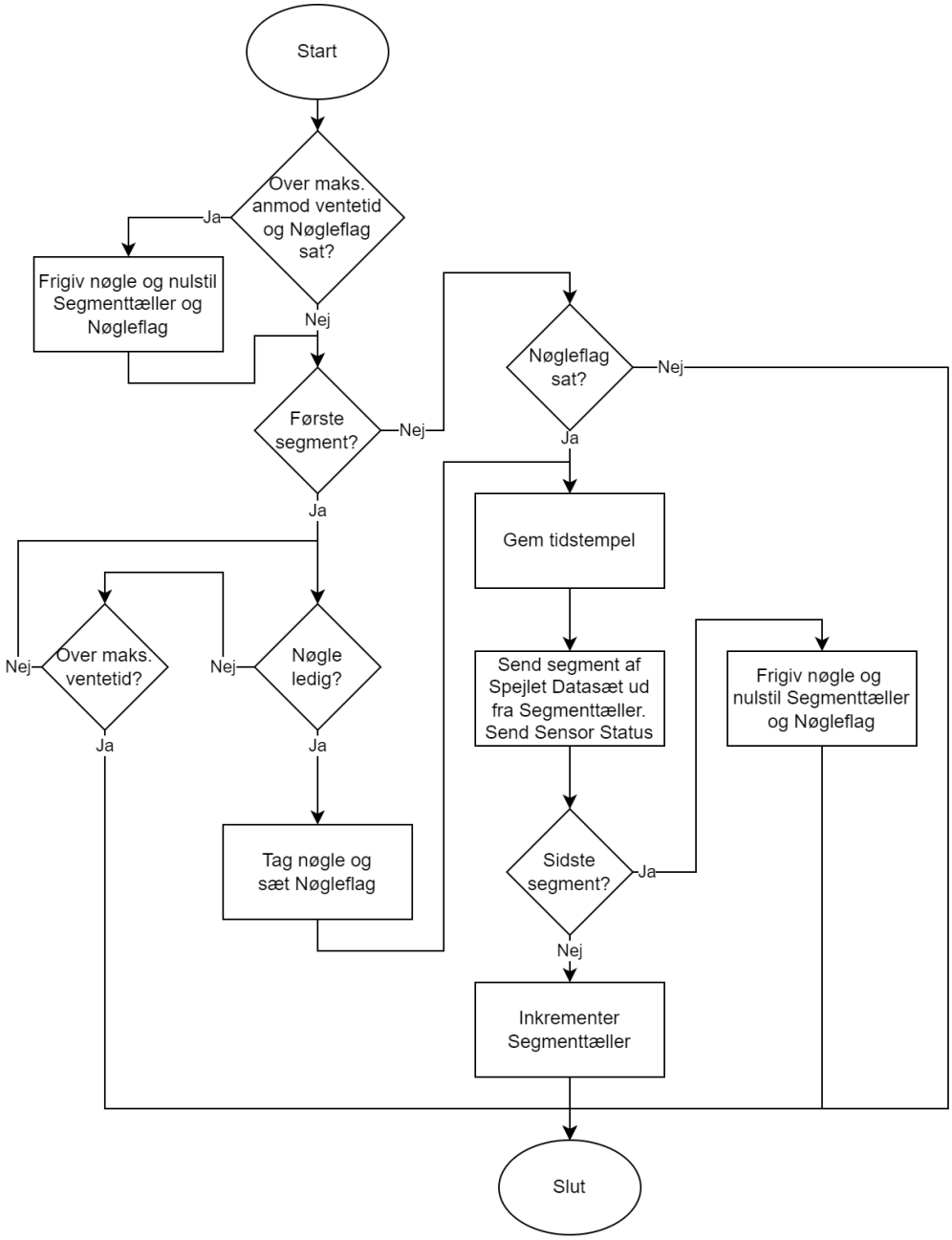
Sensor Server initialiseres med de fire sensorenheder, og der oprettes "Callback" funktioner til håndtering af "Provisioning", "Sensor Server Configuring" og "Sensor Server Communication", der kaldes af ESP BLE Mesh stakken når deres tilhørende "Events" opleves.

Efter Sensor Serveren er initialiseret, og er klar til at blive "Provisioned" af Sensor Clienten tændes en RGB LED med grønt lys på udviklingsboarded. Når Sensor Serveren er blevet "Provisioned" af Sensor Clienten ændres RGB LED farven til Blå.

Implementering

For at undgå synkroniseringsproblematikken mellem Sensor Server og Client beskrevet i Design 1 afsnit 4.2.2 sættes der en maksimal tid, der må gå mellem Sensor Serveren modtager en andmodning om segmenter.

Der oprettes en BLE Mesh Datatransmissions rutine, der tager vibrationsdata fra det Spejlet Datasæt og sender det over BLE Mesh. Figur 5.8 viser flowchart for BLE Mesh Datatransmissions rutine, der køres hver gang der modtages en "Get Sensordata" anmodning fra Server Clienten.



Figur 5.8: Flowchart for Sensor Server BLE Mesh Datatransmission.

Ved hver "Get Sensordata" anmodning undersøges det først om Nøgleflaget er sat samtidig med at der er gået over den maksimale tid mellem anmodninger. I den situation anses det som at sidste datatransmission fejlede, hvilket betyder at den nye anmodning er start af en ny datatransmission, derfor frigives nøglen og Segmenttælleren og Nøgleflaget nulstilles.

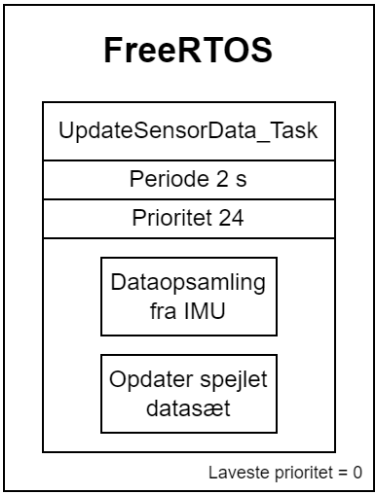
Ved anmodning om første segment af vibrationsdataene tages nøglen til det Spejlet Datasæt, hvis den er ledig, ellers ventes i op til 100 millisekunder før forsøget fejler. Efter nøglen er taget sættes Nøgleflag til at indikerer at BLE Mesh har ejerskab over nøglen, derefter sendes segment 0 fra det Spejlet Datasæt samt Sensor Status, og Segmenttælleren inkrementeres.

Ved anmodning om de resterende segmenter af vibrationsdataene afsluttes rutinen, hvis Nøgleflaget ikke er sat. Segmentet indikeret af Segmenttælleren sendes fra det Spejlede Datasæt, hvis det sidste segment er afsendt frigives nøglen efterfulgt af at Segmenttælleren og Nøgleflaget nulstilles, ellers inkrementeres Segmenttælleren.

5.2.4 RTOS

Sensor Serveren anvender FreeRTOS til håndtering af de forskellige "Tasks", der kører på SoC'en. ESP BLE Mesh har bagvedliggende "Tasks" til håndtering af "Events". Disse "Tasks" er en integreret del af ESP BLE Mesh stakken, og oprettes automatisk under opsætning og initialisering heraf.

Der oprettes en periodisk "Task" til Vibrationsmålings rutinen beskrevet på figur 5.6. "Tasken" sættes til at kører hvert andet sekund, og prioriteten sættes til 24, hvor maks. er 25, for at sikre at målingen af vibrationsdataene sker med et konstant interval, ved at undgå at "Tasken" bliver unødigt afbrudt af andre "Tasks".



Figur 5.9: Sensor Server FreeRTOS task.

Figur 5.9 viser navn, periode, prioritet og funktionalitet for den "Task", der kører på Sensor Serveren.

5.3 Sensor Client

Sensor Clientens tre hovedfunktionaliteter består af modtagelse af sensordata over BLE Mesh netværk, internetforbindelse gennem Wi-Fi og opdatering af Tidsstempel, samt transmission af Tidsstempel og vibrationsdata over MQTT til AWS.

Datahåndteringen, implementering af BLE Mesh kommunikation, implementering af Wi-Fi funktionaliteter, implementering af AWS MQTT kommunikation og FreeRTOS "Tasks" tilhørende Sensor Clienten gennemgås.

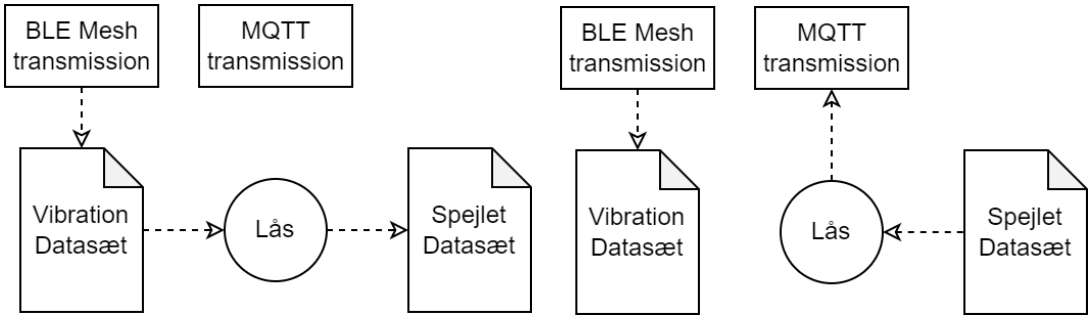
5.3.1 Datahåndtering

Sensor Clienten har samme problematik med korruption af Vibrations Datasæt som Sensor Serveren beskrevet i afsnit 5.2.1. På samme måde anvendes der et Spejlet Datasæt, der låses med en nøgle.

Implementering

Når Sensor Clienten over BLE Mesh har modtaget et komplet Vibrations Datasæt tages nøglen. Derefter opdateres det Spejlet Datasæt, og nøglen frigives.

Før MQTT transmission af det Spejlet Datasæt tages nøglen. Efter transmissionen er færdig frigives nøglen.



(a) BLE Mesh har nøglen. ***(b)*** MQTT har nøglen.

Figur 5.10: Anvendelse af det låste Spejlet Datasæt.

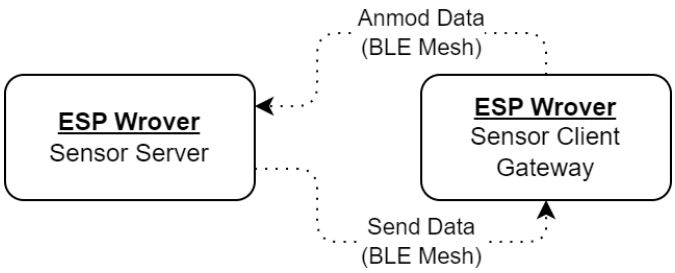
Figur 5.10 illustrerer anvendelsen af det låste Spejlet Datasæt, hvor de stiplede pile indikerer dataoverførsel. På figur 5.10a har BLE Mesh nøglen, hvilket gør det muligt at opdaterer det Spejlet Datasæt med data modtaget over BLE Mesh. På figur 5.10b har MQTT nøglen, hvilket gør det muligt sende det Spejlet Datasæt over MQTT.

Maks. ventetid for MQTT transmission sættes til 100 ms ud fra opdateringstiden af det Spejlet Datasæt inklusiv en ekstra margin. Maks. ventetid for opdatering af det Spejlet Datasæt med vibrationsdata modtaget over BLE Mesh sættes til 1 s udfra transmissionstiden af datasættet over MQTT inklusiv en ekstra margin.

Implementering af det låste Spejlet Datasæt gøres på samme måde som for Sensor Server beskrevet i afsnit 5.2.1. FreeRTOS stakken anvendes til at oprette en "Mutex" som lås, hvorefter FreeRTOS funktioner anvendes til at tage og frigive den.

5.3.2 BLE Mesh

Afsnittet gennemgår implementering af BLE Mesh kommunikation på Sensor Clienten, hvor initialisering af Sensor Server og BLE Mesh Datatransmission rutinen gennemgås.



Figur 5.11: BLE Mesh kommunikationsdiagram.

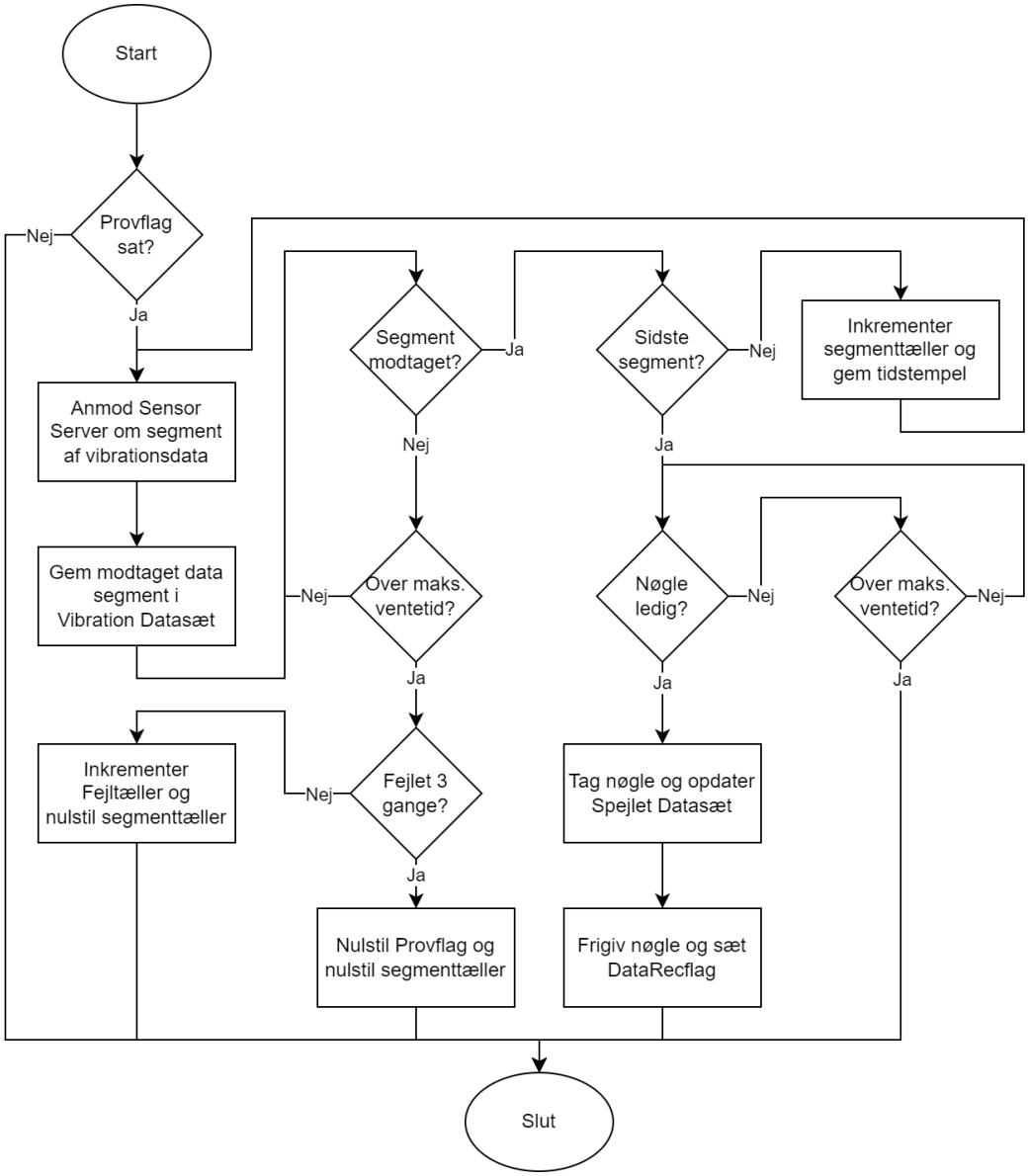
Figur 5.11 viser et kommunikationsdiagram for BLE Mesh delsystemet bestående af Sensor Server (ESP Wrover) og Sensor Client (ESP Wrover). Sensor Client anmoder Sensor Server om et segment af Sensordataene over BLE Mesh, hvorefter Sensor Server sender datasegmentet. Dette gentages indtil alle segmenter er sendt.

Sensor Client initialiseres, og der oprettes "Callback" funktioner til håndtering af "Provisioning", "Sensor Client Configuring" og "Sensor Client Communication", der kaldes af ESP BLE Mesh stakken når deres tilhørende "Events" opleves.

Efter Sensor Clienten er initialiseret, og er klar til at "Provision" Sensor Serveren tændes en RGB LED med grønt lys på udviklingsboarded. Når Sensor Serveren er blevet "Provisioned" af Sensor Clienten ændres RGB LED farven til Blå.

Implementering

Figur 5.12 viser flowchart for BLE Mesh Datatransmissions rutine, hvor Sensor Clienten sender en "Get Sensordata" admodning til Sensor Serveren, hvis Sensor Serveren er "Provisioned".



Figur 5.12: Flowchart for Server Client BLE Mesh Datatransmission.

Hver gang BLE Mesh Datatransmissions rutinen køres undersøges det først om Sensor Serveren er "Provisoned". Herefter sendes en "Get Sensordata" anmodning til Sensor Serveren, og der ventes på at modtage vibrationsdata segmentet i maksimum 6 s før datatransmissionen fejler.

Hvis datatransmissionen har fejlet tre gange i træk anses Sensor Serveren som fejlet og ikke "Provisioned", derfor nulstilles Provflag og segmenttælleren.

Efter modtagelse af segmentet undersøges det om det modtagede segment er det sidste. Hvis det ikke er det sidste segment tælles segmenttælleren op og der anmodes om et nyt segment vibrationsdata.

Efter modtagelse af det sidste segment tages nøglen til det Spejlet Datasæt, hvis den er ledig, ellers ventes i op til 1 s før forsøget fejler (bestemt i afsnit 5.3.1). Efter nøglen er taget opdateres det Spejlet Datasæt med dataene modtaget over BLE Mesh, nøglen frigives og BLE Mesh Datatransmissions rutinen afsluttes.

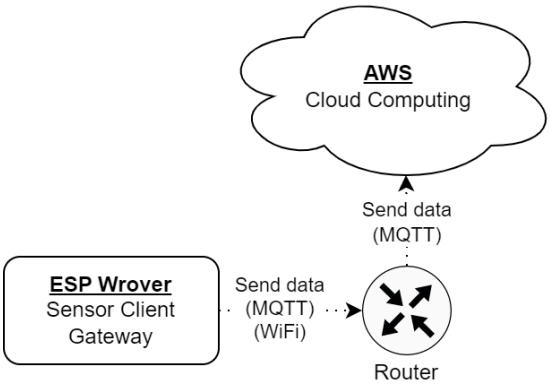
5.3.3 Wi-Fi

ESP Wi-Fi stakken initialiseres og anvendes til at oprette forbindelse til et kendt Wi-Fi netværk med internetforbindelse. Efter der er oprettet forbindelse til internettet anvendes funktioner beskrevet i ESP Wi-Fi stakken til at forbinde til NTP Serveren "pool.ntp.org", hvor "Unix" Tidsstemplet aflæses og gemmes.

Herefter tælles "Unix" Tidsstemplet op i forhold til den interne "Clock" på Sensor Clienten. For at sikre at "Unix" Tidsstemplet ikke skrider over tid kan der periodisk oprettes forbindelse til NTP Serveren og udlæses det korrekte "Unix" Tidsstempl.

5.3.4 AWS MQTT

AWS tilbyder en færdig stak til ESP Wrover med implementering af MQTT kommunikation til AWS, der anvender FreeRTOS til implementering heraf.



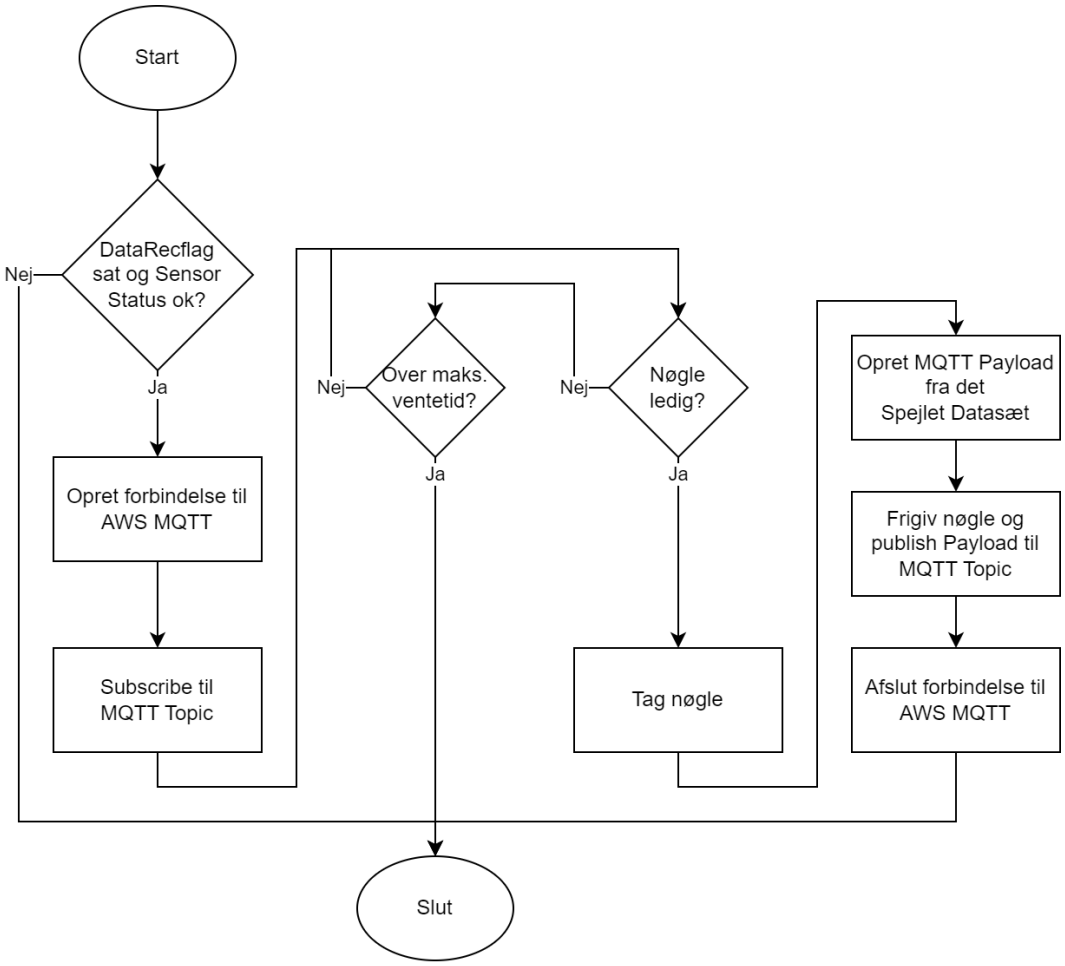
Figur 5.13: Cloud Computing kommunikationsdiagram.

Figur 5.13 et kommunikationsdiagram for Cloud Computing delsystemet bestående af Sensor Client (ESP Wrover), der over Wi-Fi sender vibrationsdataene med MQTT til AWS.

AWS MQTT stakken initialiseres med autentifikations nøgler og certifikater beskrevet i afsnit 4.3.2. Der oprettes en "Topic" kaldet "IIoT/Predictive/1/Data", og MQTT forbindelsen knyttes til "Endpoint" for AWS IoT Core.

Implementering

Figur 5.14 viser flowchart for AWS MQTT Datatransmissions rutine, der sender det Spejlet Datasæt til AWS Cloud hvergang der modtages et nyt Vibrations Datasæt fra Sensor Serveren.



Figur 5.14: Flowchart for Server Client MQTT Datatransmission.

Hver gang AWS MQTT Datatransmissions rutinen køres undersøges det først om det Spejlet Datasæt er opdateret med et nyt Vibrations Datasæt modtaget over BLE Mesh. Herefter kontrolleres Sensor Status for at undersøge om det modtagede vibrationsdata er valid.

Når den modtagede vibrationsdata er valid oprettes der forbindelse til AWS MQTT, hvorefter der "Subscribes" til MQTT "Topic". Herefter tages nøglen til det Spejlet Datasæt, hvis den er ledig, ellers ventes i op til 100 ms før forsøget fejler (bestemt i afsnit 5.3.1). Efter nøglen er taget oprettes MQTT "Payload" med vibrationsdataene fra det Spejlet Datasæt, og nøglen frigives. Herefter publiseres "Payload" til MQTT "Topic", forbindelsen til AWS afsluttes og AWS MQTT Datatransmissions rutinen afsluttes.

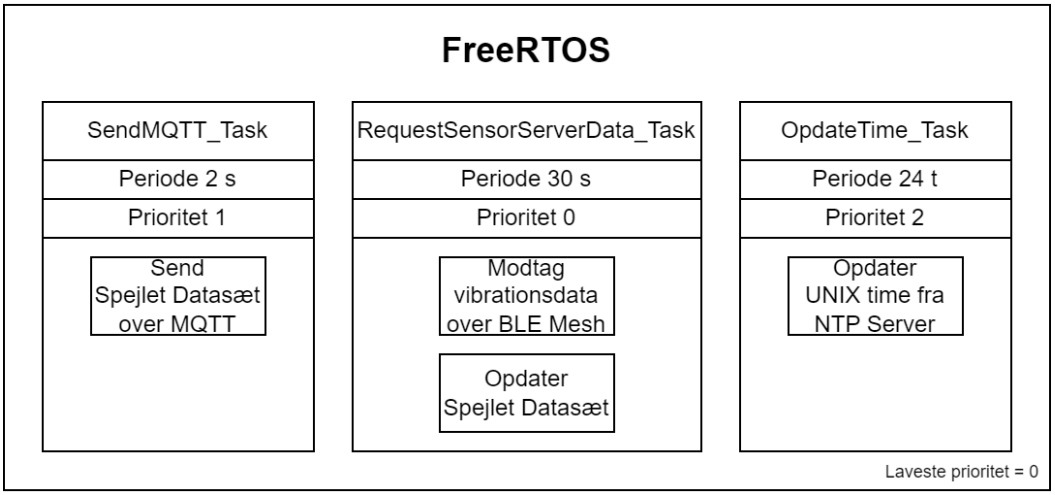
5.3.5 RTOS

Sensor Clienten anvender lige som Sensor Serveren FreeRTOS til håndtering af de forskellige "Tasks", der kører på SoC'en. ESP BLE Mesh og AWS MQTT har bagvedliggende "Tasks" til håndtering af "Events". Disse "Tasks" er en integreret del af stakkene, og oprettes automatisk under opsætning og initialisering heraf. Prioritet for de forskellige "Tasks" er ikke kritisk, og er derfor sat i forhold til deres eksekveringstid, hvor længeste eksekveringstid får lavest prioritet.

Der oprettes en periodisk "Task" til BLE Mesh Datatransmissions rutinen beskrevet på figur 5.12. "Tasken" sættes til at kører hvert 30. sekund, og prioriteten sættes til 0.

Der oprettes en periodisk "Task" til AWS MQTT Datatransmissions rutinen beskrevet på figur 5.14. "Tasken" sættes til at kører hvert andet sekund, og prioriteten sættes til 1.

Der oprettes en periodisk "Task" til at oprette forbinde til NTP Serveren "pool.ntp.org" for at opdaterer "Unix" Tidsstemplet. "Tasken" sættes til at kører hver 24. time, og prioriteten sættes til 2.



Figur 5.15: Sensor Client FreeRTOS tasks.

Figur 5.15 viser navn, periode, prioritet og funktionalitet for de "Tasks", der kører på Sensor Clienten.

5.4 Cloud Computing

Afsnittet gennemgår implementering af Cloud Computing i AWS, hvor initialisering af IoT Core, S3, SNS og Lambda, samt Lambda CbM rutinen gennemgås.

I IoT Core oprettes der en ny enhed til Sensor Clienten, og samtidig auto-genereres der autentificerings nøgler og certifikater. Der oprettes en regel, der videresender alle MQTT beskeder modtaget på "Topic" "IIoT/Predictive/1/Data" til Lambda, hvor dette registreres som et "Event".

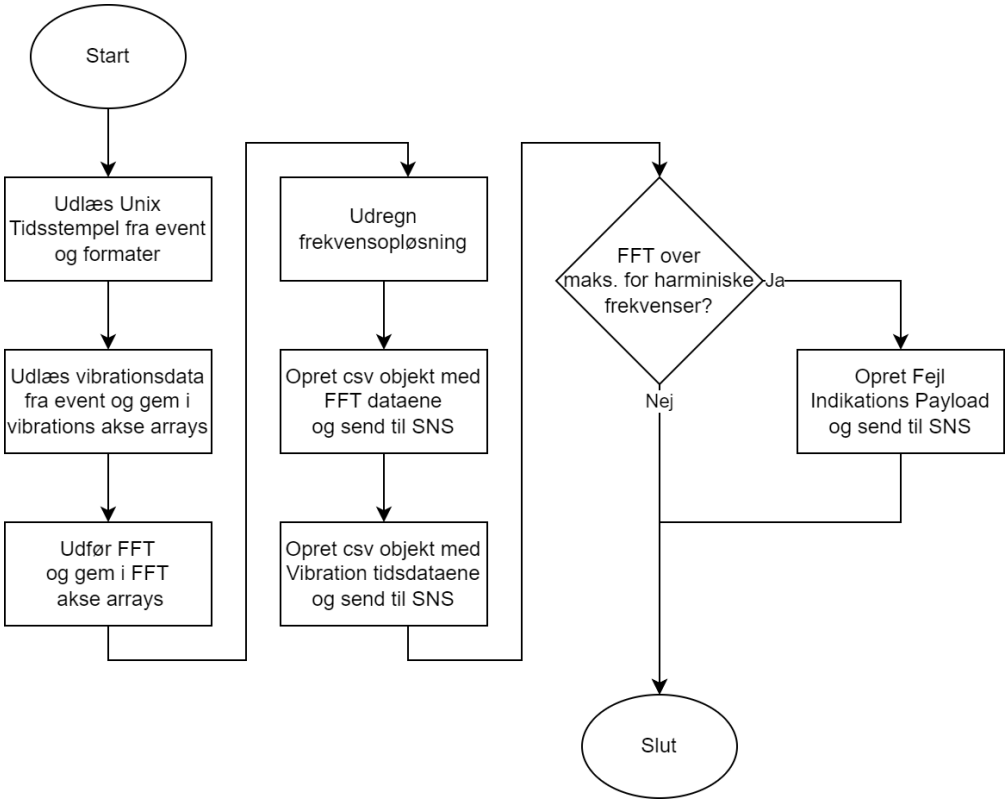
I S3 oprettes der en ny "Bucket" til opbevaring af vibrationsdataene i tids- og frekvensdomænet. Der oprettes en "Lifecycle Rule", der fjerner vibrationsdata ældre end 30 dage. Dette sørger for at det kun er relevante vibrationsdata, der gemmes og at datamængden i den oprettede "Bucket" holdes til et minimuml.

I SNS oprettes der en "Topic", der håndterer videresendelse af Fejl Indikations "Payload" til serviceperson over E-mail.

I Lambda oprettes der en "Function" med programmeringssproget Python, der køres hver gang der registreres et "Event" for modtagelse af ny vibrationsdata fra IoT Core. Der anvendes biblioteket "Datetime" til formatering af "Unix" Tidsstempel, "Numpy" til udførsel af FFT og "Boto3" til at gemme vibrationsdataene i S3 og sende Fejl Indikations "Payload" til SNS.

Implementering

Figur 5.16 viser flowchart for Lambda CbM rutine, der køres hver gang der modtages ny vibrationsdata fra IoT Core. Dataene fra MQTT "Payload" udlæses og opsplittes, og der FFT udføres på vibrationsdataene. Vibrationsdataene for tids- og frekvensdomæne gemmes i S3. FFT dataene sammenlignes med maksimale værdier, og ved fejl sendes der en Fejl Indikations "Payload" til SNS.



Figur 5.16: Flowchart for Lambda CbM rutine

Hver gang Lambda CbM rutinen køres udlæses "Unix" Tidsstempelet fra MQTT "Payload", og denne formateres til "dag/måned/år time:minut:sekund" til at sende i Fejl Indikations "Payload", samt "dag_måned_år_time_minut_sekund" til oprettelse af mappe med vibrations data for tids- og frekvensdomæne i S3.

Vibrationsdataene udlæses fra MQTT "Payload", og opsplittes til et array per akse med vibrationsdataene 0 til 127. Der udføres FFT på de tre array med vibrationsdata, og de reelle værdier gemmes i et array per akse. Størrelsen ved de forskellige frekvenser udregnes, skalleres til g og gemmes i et array per akse. Frekvensopløsningen udregnes ud fra samplingsfrekvensen på 125 Hz og antal målinger på 128.

Der oprettes et objekt med csv formatering til de skallerede FFT data, hvor hver række indeholde "Frekvens;X-akse;Y-akse;Z-akse". Csv objektet med de skallerede FFT data gemmes i S3 "Bucket" i en mappe ud fra "Unix" Tidsstemplet.

Array med vibrationsdataene skalleres til g, og der oprettes et objekt med csv formatering, hvor hver række indeholde "Tid;X-akse;Y-akse;Z-akse". Csv objektet med de skallerede vibrations tidsdata gemmes i S3 "Bucket" i en mappe ud fra "Unix" Tidsstemplet.

De skallerede FFT data sammenlignes med den maksimale målte g-påvirkning inklusiv den ekstra margin for hver akse i frekvensspændende ± 2 Hz omkring de fire harmoniske frekvenser. Hvis en af sammenligningerne er uden for specifikation oprettes en Fejl Indikations "Payload" som specificeret i Design afsnit 4.3.3.

Accepttest 6

Der er formuleret to test til validering om prototype IIoT prædiktiv vedligeholdelsessystemet opfylder de funktionelle og tekniske krav opstillet i kravspecifikationen i kapitel 3. Hoved punkterne fra målejournal D med de funktionelle krav, og målejournal E med det tekniske krav vil fremgå i følgende kapitel.

6.1 Funktionelle krav

Målejournal D gennemgår test af de funktionelle krav. De funktionelle krav vist i tabel 6.1 testes i en samlet test.

Tabel 6.1: Oversigt over funktionelle krav.

Krav	Beskrivelse
F.1	Vibrationsmåling
F.2	Intern kommunikation BLE Mesh netværk
F.3	Ekstern kommunikation Wi-Fi internetforbindelse
F.4	Fejl Indikations Notifikation

6.1.1 Fremgangsmåde

1. Tænd PSU og indstil til 25 VDC 5 A
2. Sluk PSU output
3. Tilslut PSU plus til brun ledning fra aktuator
4. Tilslut PSU minus til blå ledning fra aktuator
5. Tænd for Sensor Client og vent på RGB LED lyser grøn
6. Når RGB LED på Sensor Clienten lyser grøn har den internetforbindelse over Wi-Fi, og overholder derfor krav F.3
7. Tænd for Sensor Server og vent på RGB LED på begge enheder bliver blå
8. Når RGB LED på begge enheder lyser blå har Sensor Clienten oprettet et BLE Mesh netværk med Sensor Serveren, og overholder derfor krav F.2
9. Åben browser på Computer og tilgå AWS S3 "Bucket"
10. Tænd PSU output, så aktuatoren kører ud
11. Vent 3 s
12. Sluk PSU output, så aktuatoren stopper
13. Vent på at der er genereret en ny mappe i AWS S3 "Bucket" med tidsstempel indeholdende en csv fil med vibrations tidsdata og en med FFT data
14. Gem de genererede csv filer med FFT data og vibrations tidsdata fra AWS S3 "Bucket"
15. Valider at de generede filer indeholder data der ikke er nul, hvilket betyder at der er målt vibrationsdata, og overholder derfor krav F.1
16. Tænd PSU output, så aktuatoren kører ud
17. Påvirk den elektriske lineær aktuator i minimum 3 s mens den kører ved at ryste den under vibrationsmålingen
18. Sluk PSU output, så aktuatoren stopper
19. Vent på at der er modtaget en Fejl Indikations E-mail, hvilket betyder at krav F.4 overholders

6.1.2 Konklusion

I testen af de funktionelle krav kunne alle trin gennemgås, hvilket betyder at prototype IIoT prædiktiv vedligeholdelsessystemet overholder alle de funktionelle krav.

Tabel 6.2: Oversigt over testede funktionelle krav.

Krav	Forklaring	✓/✗
F.1	Vibrationsmåling	✓
F.2	Intern kommunikation BLE Mesh netværk	✓
F.3	Ekstern kommunikation Wi-Fi internetforbindelse	✓
F.4	Fejl Indikations Notifikation	✓

6.2 Tekniske krav

Målejournal E gennemgår test af det tekniske krav. Det tekniske krav vist i tabel 6.1 testes i en samlet test.

Tabel 6.3: Oversigt over tekniske krav til tests.

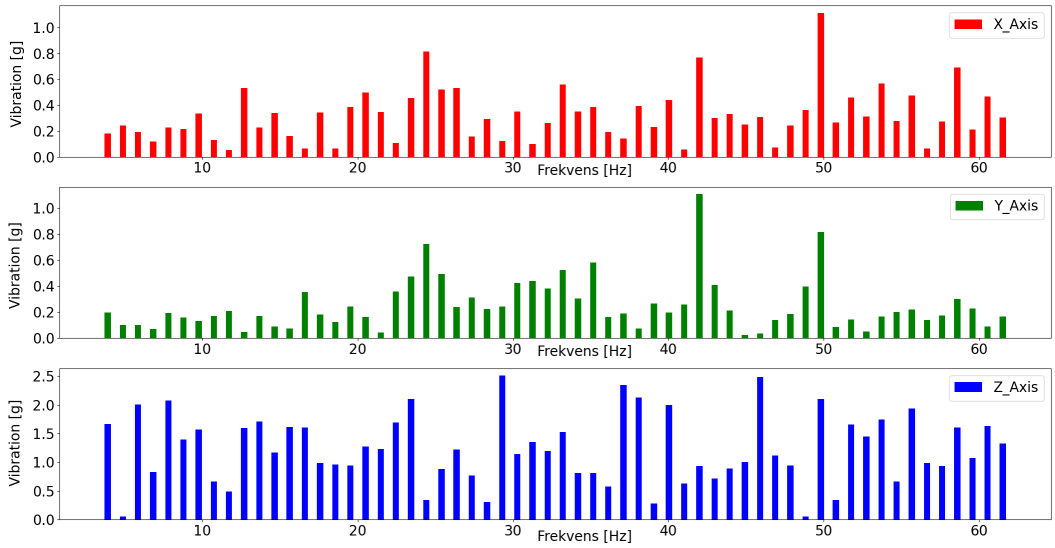
Krav	Beskrivelse	Værdi
T.1	Vibrationsmålings	13,85 – 55,40 Hz

6.2.1 Fremgangsmåde

1. Tænd PSU og indstil til 25 VDC 5 A
2. Sluk PSU output
3. Tilslut PSU plus til brun ledning fra aktuator
4. Tilslut PSU minus til blå ledning fra aktuator
5. Tænd for Sensor Client og vent på RGB LED lyser grøn
6. Tænd for Sensor Server og vent på RGB LED på begge enheder bliver blå
7. Åben browser på Computer og tilgå AWS S3 "Bucket"
8. Tænd PSU output, så aktuatoren kører ud
9. Vent 3 s
10. Sluk PSU output, så aktuatoren stopper
11. Gem den genererede csv filer med FFT data fra AWS S3 "Bucket"

Resultatbehandling

FFT dataene fra AWS S3 "Bucket" er vist i målejournal E, og der laves en graf for hver akse, hvor grafernes X-akse viser frekvens i Hz og Y-aksen g-påvirkning i g, se figur 6.1.



Figur 6.1: Graf med FFT data fra AWS S3 Bucket.

6.2.2 Konklusion

I tabel E.3 i målejournal E aflæses den laveste målte frekvens til 3,91 Hz, og i tabel E.4 i målejournal E aflæses den højeste målte frekvens til 61,52 Hz, hvilket også kan ses på figur 6.1.

Det ønskede frekvensspænd fra krav T.1 er indeholdt i det målte frekvensspænd, hvilket betyder at prototype IIoT prædiktiv vedligeholdelsessystemet overholder det tekniske krav.

Tabel 6.4: Oversigt over de testede tekniske krav.

Krav	Beskrivelse	Værdi	✓/✗
T.1	Vibrationsmålings	13,85 – 55,40 Hz	✓

6.3 Konklusion af accepttest

Tabel 6.5 viser de funktionelle krav og det tekniske krav, samt om prototype IIoT prædiktiv vedligeholdelsessystemet overholder kravene.

Tabel 6.5: Samlet tabel over testede funktionelle- og tekniske krav.

Funktionelle krav			
Krav.nr	Beskrivelse		Opfyldt
F.1	Vibrationsmåling		✓
F.2	Intern kommunikation BLE Mesh netværk		✓
F.3	Ekstern kommunikation Wi-Fi internetforbindelse		✓
F.4	Fejl Indikations Notifikation		✓
Tekniske krav			
Krav	Beskrivelse	Værdi	Opfyldt
T.1	Vibrationsmålings	13,85 – 55,40 Hz	✓

Ud fra tabel 6.5 konkluderes det at prototype IIoT prædiktiv vedligeholdelsessystemet overholder alle de funktionelle og tekniske krav fra Kravspecifikationen kapitel 3.

Konklusion 7

Ud fra analysen af motivationen for implementering af et IIoT prædiktiv vedligeholdelsessystem, samt anvendelige kommunikationsteknologier og overvejelser til Cyper Secutity er der opstillet et ideelt IIoT prædiktiv vedligeholdelsessystem.

Der afgrænses til udviklingen af et prototype IIoT prædiktiv vedligeholdelsessystem bestående af en Sensor Server, der måler vibrationsdata. Dataene transmitteres over BLE Mesh til en Sensor Client, der transmittere dataene videre til AWS Cloud Computing over Wi-Fi. Ud fra prototype IIoT prædiktiv vedligeholdelsessystemet er der blevet udarbejdet en kravspecifikation bestående af fire funktionelle krav og et teknisk krav.

Prototype IIoT prædiktiv vedligeholdelsessystemet er opdelt i tre delsystemer bestående af Dataopsamling, BLE Mesh og Cloud Computing. Dataopsamling delsystemet omhandler måling af vibrationsdata, BLE Mesh delsystemet omhandler den interne kommunikation og Cloud Computing delsystemet omhandler den eksterne kommunikation, samt udførelse af CbM i AWS.

Der er blevet udført en samlet accepttest af alle delsystemerne i prototype IIoT prædiktiv vedligeholdelsessystemet. Accepttesten består af test af de fire funktionelle krav, og test af det tekniske krav. Ud fra accepttesten konkluderes det at prototype IIoT prædiktiv vedligeholdelsessystemet overholder alle de funktionelle og tekniske krav.

Design

Dataopsamling består af Sensor Serveren og Vibrationssensoren, og omhandler måling af vibrationsdata for den elektriske lineær aktuator, når den er i bevægelse. Samplingsfrekvens på 125 Hz, antal målinger på 128 og datamængde på 768 byte for vibrationsmålingen bestemmes i forhold til krav T.1.

BLE Mesh består af Sensor Serveren og Sensor Clienten, og omhandler den interne kommunikation over BLE Mesh. Sensor Serveren opsættes med fire sensorenheder bestående af vibrationsdata for X-, Y- og Z-akse og Sensor Status til indikation af validiteten af vibrationsdataene. Vibrationsdataene for de tre akser opsplittes til fire datatransmissioner bestående af 65-byte for hver akse, hvor 1-byte anvendes til at indikerer segmentnummer. Den samlede Transport PDU for hver af de fire datatransmissioner består af 208-byte.

Cloud Computing består af Sensor Clienten og AWS Cloud Computing, og omhandler den eksterne kommunikation over MQTT, samt udførelse af CbM. Sensor Clienten anvender Wi-Fi som tilslutning til internettet, hvor et "Unix" Tidsstempel kan udlæses fra en NTP Server. MQTT anvendes til at sende vibrationsdataene til AWS, hvor der oprettes en "Payload" bestående af "Unix" Tidsstemplet og vibrationsdataene for de tre akser. Vibrationsdataene grupperes for hver akse, og indikeres med f.eks. X0 til X127. For at kunne udføre CbM transformeres vibrationsdataene fra tidsdomænet til frekvensdomænet ved hjælp af FFT. Slid på f.eks. kuglelejer og gear vil fremstå som en forhøjet amplitude omkring de harmoniske frekvenser for den elektriske lineær aktuators RPM. Amplituden for frekvensspændene på ± 2 Hz omkring de harmoniske frekvenser sammenlignes med maks. værdier bestemt ved test for en velfungerende aktuator inklusiv en ekstra margin. I tilfælde, hvor vibrationsdataene ligger uden for specifikation oprettes der en Fejl Indikations E-mail bestående af enhedsnavn, Tidsstempel for vibrationsmåling, samt de kombinationer af akser og harmoniske frekvenser, der er uden for specifikation. Til Cloud Computing anvendes AWS, hvor IoT Core fungerer som en "Message Broker" til MQTT datatransmissionerne, Lambda fungerer som en "Event-driven" BaaS til udførelse af CbM, S3 fungerer som "Object Storage" til opbevaring af vibrationsdataene før og efter FFT og SNS fungerer som "Notifications Service" til at sende Fejl Indikations E-mailen.

Implementering

Sensor Server har to hovedfunktionaliteter bestående af Dataopsamling og datatransmission over BLE Mesh, og implementeres på et ESP Wrover udviklingsboard. Datahåndtering sikre vibrationsdataene for korruption ved anvendelse af et låst Spejlet Datasæt, der implementeres med FreeRTOS stakkens "Mutex" som lås. BMX160 IMU anvendes som Vibrationssensor, hvor Sensor Serveren over I^2C anmoder om sensordata. Forsyningen til den elektriske lineær aktuator tilsluttes gennem et målekredsløb til et digitalt input på udviklingsboarded, hvor inputtet bliver højt når aktuatoren er i bevægelse. Der oprettes en Vibrationsmålings rutine til at udlæse 128 målinger fra Vibrationssensoren med 7,648 ms mellemrum, hvis aktuatoren er i bevægelse og der er gået mere end 20 s siden sidste måling, hvorefter vibrationsdataene gemmes i det Spejlet Datasæt. ESP BLE Mesh stakken anvendes, hvor der oprettes en BLE Mesh Datatranmissions rutine til at sende vibrationsdataene fra det Spejlet Datasæt, når Server Sensoren modtaget en anmodning om sensordata fra Sensor Clienten. FreeRTOS stakken anvendes til at oprette en periodisk "Task" til Vibrationsmålings rutinen, der køres hvert andet sekundt.

Sensor Client har tre hovedfunktionaliteter bestående af modtagelse af sensordata over BLE Mesh, internetforbindelse gennem Wi-Fi med opdatering af Tidsstempel og transmission af Tidsstempel og vibrationsdata over MQTT til AWS, og implementeres på et ESP Wrover udviklingsboard. Datahåndtering foregår på samme måde som på Sensor Serveren. ESP BLE Mesh stakken anvendes, hvor der oprettes en BLE Mesh Datatranmissions rutine til at anmode Sensor Serveren om sensordata, når denne er "Provisioned", hvorefter vibrationsdataene gemmes i det Spejlet Datasæt. FreeRTOS stakken anvendes til at oprette en periodisk "Task" til BLE Mesh Datatranmissions rutinen, der køres hvert 30. sekundt. ESP Wi-Fi stakken anvendes, hvor der tilsluttes et kendt Wi-Fi og "Unix" Tidsstemplet udlæses fra en NTP Server. FreeRTOS stakken anvendes til at oprette en periodisk "Task" til opdatering af Tidsstemplet, der køres hver 24. time. AWS MQTT stakken anvendes, hvor der oprettes en AWS MQTT Datatransmissions rutine til at "Publish" vibrationsdataene fra det Spejlet Datasæt med "Topic" "IIoT/Predictive/1/Data" til AWS, hvis der er modtaget en valid vibrationsmåling fra Sensor Serveren. FreeRTOS stakken anvendes til at oprette en periodisk "Task" til AWS MQTT Datatransmissions rutinen, der køres hvert andet sekundt.

Cloud Computing har tre hovedfunktionaliteter bestående af modtagelse af Tidsstempel og vibrationsdata over MQTT, udførsel af CbM og fejl indikation ved målinger uden for specifikation, og implementeres med AWS. IoT Core anvendes til MQTT datatransmission, hvor der genereres autentficerings nøgler og certifikater, og oprettes en regel til at videresende vibrationsdataene til Lambda. S3 anvendes til at opbevare vibrationsdataene til senere brug, hvor der oprettes en "Bucket", der fjerner datasæt ældre end 30 dage. SNS anvendes til transmission af Fejl Indikations E-mail, hvor der oprettes en "Topic", der sender E-mailen til en serviceperson. Lambda anvendes, hvor der oprettes en rutine til udførelse af FFT på vibrationsdataene, gemme vibrationsdataene for tids- og frekvensdomæne i S3, udføre CbM ved at sammenligne FFT dataene med maksimale værdier og oprette en Fejl Indikations "Payload" ved målinger uden for specifikation, der køres hver gang der modtages nyt vibrationsdata.

Diskussion 8

Følgende kapitel omhandler diskussion af og eventuelle forbedringer til prototype IIoT prædiktiv vedligeholdelsessystemet.

Matter er efter udarbejdelse af analysen kapitel 2 blevet udgivet med Matter 1.0 standarden d. 4. oktober 2022. Bl.a. Amazon og Google introduceret integrering af Matter 1.0 standarden, og der er annonceret en række smart produkter fra bl.a. Bosch og Nanoleaf, der anvender Matter 1.0.

Dataopsamling omhandler kun vibrationsmåling, idet denne har den største datamængde, og derved er mest kompleks at implementere p.g.a. den begrænsede maksimale Transport PDU for BLE Mesh. For at tilføje temperatur og strømtræksmåling skal der i dataopsplitnings design 1 tilføjes én sensorenhed til måling af temperatur, og én til måling af strømtræk for den elektriske lineær aktuator. Temperaturmåling vil kunne implementeres simpelt med f.eks. en PT100, NTC eller anden Resistance Temperature Detector (RTD). Strømtræk vil kunne implementeres simpelt ved at måle spændingsfaldet over en modstand, der sidder i serie med forsyningen til aktuatoren, eller ved et eksternt modul, der kommunikeres med over f.eks. I^2C. Accelerometret måler vibrationsdata for X-, Y- og Z-aksen, hvilket betyder at vibrationsdataene vil være afhængige af IMU'ens rotation. Grænseværdierne, der anvendes til CbM, for Z-aksen har større amplitude end X- og Y-aksen, hvilket betyder at en rotation af IMU'en vil resultere i systemfejl. Herved vil det være nødvendigt at bestemme nye grænseværdier for alle applikationer, hvor IMU'en har forskellig rotation. I applikationer, hvor rotationen af aktuatoren er dynamisk under kørsel, vil dette også kunne give problemer i forhold til udførsel af CbM. Gyroskopet i BMX160 IMU'en vil kunne anvendes til at transformere dataene fra accelerometret, så de bliver uafhængige af rotationen af aktuatoren.

BLE Mesh er ikke beregnet til at overføre den datamængde, der er nødvendig for udførelse af CbM ud fra vibrationsdata. Dataoverførelsen er begrænset til 127-byte per sensorenhed, og 384-byte for den totale Transport PDU. Derved er det nødvendigt at opsplitte vibrationsdataene i flere datatransmissioner, hvilket øger risikoen for fejl under transmission, samt transmissionstiden af det samlede Vibrations Datasæt, og derfor stiller større krav til data- og fejlhåndtering. Sensor Clienten sender hvert 30. sekundt en anmodning om sensordataene fra Sensor Serveren, hvilket resulterer i at der er et "Delay" mellem vibrationsdataene bliver målt, og at de modtages af Sensor Clienten. Ved at sænke tiden mellem anmodninger om vibrationsdataene vil reaktionstiden af systemet kunne forbedres. En for kort tid mellem anmodninger kan resultere i at BLE Mesh netværket "Floddes", hvor problemet vil førøges i forhold til antallet af Sensor Servere, der er tilsluttet til samme Sensor Client. Den værste tænkelige reaktionstid for BLE Mesh netværket vil være når der udføres en vibrationsmåling imens BLE Mesh Datatransmissionen er igangværende. Her vil dataene fra vibrationsmålingen først sendes ved næste anmodning, der modtages efter ca. 35 s på grund af periodetiden for BLE Mesh Datatransmissions "Tasken" på Sensor Clienten og transmissionstiden for dataene. Med denne implementering af BLE Mesh er det ikke muligt for Sensor Serveren at sende vibrationsdataene uden en anmodning fra Sensor Clienten. For at løse problematikken med dataopsplitning og reaktionstid vil det være nødvendigt at undersøge et alternativt Mesh netværk til BLE Mesh, der har mulighed for transmission af "store" datamængder, og hvor Sensor Serveren kan sende data uden en anmodning. Herunder kunne det undersøges om Matter 1.0 standarden har samme restriktioner som BLE Mesh i forhold til datamængder og sending af sensordata uden anmodning fra Client.

Cloud Computing anvendes til at udføre en simpel CbM, hvor der indikeres, hvis de målte data er uden for specifikation. Der indikeres ikke, hvad der er skyld i fejlen, eller hvor kritisk fejlen er. Det vil være nødvendigt at udføre en lang række test for at måle, hvilke harmoniske frekvenser forskellige fejl fremtræder som, samt hvor kritisk fejlen er ved forskellige amplituder. Herved kunne der tilføjes en række stadier, hvor der ved målinger uden for specifikation i et bestemt spænd for en bestemt harmonisk frekvens indikeres en specifik fejl samt, hvor kritisk denne er. En anden tilgang til CbM kunne være Machine Learning, hvor der på samme måde laves en række målinger på velfungerende aktuatorer, samt aktuatorer med kendte fejl. Herefter trænes en Machine Learning model på de målte data, hvorefter modellen vil kunne anvendes til at udføre CbM.

Reaktionstid for prototypen kommer primært fra BLE Mesh som beskrevet tidligere (35 s). Reaktionstiden for MQTT datatransmission og udførelsen af CbM anslås til et par sekunder. Den værste totale reaktionstid vil derfor være ca. 37 s. Dette vil i de fleste tilfælde være tilstrækkeligt til udførsel af CbM, idet der ved prædiktiv vedligeholdelse indikeres påbegyndende fejl, og derved ikke kræver hurtig reaktion af systemet.

Fejl Indikation for prototypen er integreret med en Fejl Indikations E-mail, der sendes til en serviceperson. Dette vil eventuelt kunne suppleres med en statuslampe tilhørende den elektriske lineær aktuator, der skifter farve alt efter om service er nødvendigt. Dette vil simplificere servicepersonens arbejde med at identificere maskiner, der har brug for service, samtidig vil det give et ekstra sikkerhedsnet i tilfælde af f.eks. fejl med E-mail servere. Dette vil kunne implementeres på Sensor Serverene, hvor de også opsættes som en On/Off Server, og hvor Sensor Clienten også opsættes som On/Off Client. Derved har Sensor Serverene en indikationslampe, der kan tændes og slukkes af Sensor Clienten.

Litteratur

[1] Dansk Industri. Hvad er iot? https://www.danskindustri.dk/vi-radgiver-dig/forretningsudvikling/digitalisering-og-innovation/grib-nye-teknologier/iot/. (Accessed on 09/06/2022).

[2] Keith R. Mobley. *An Introduction to Predictive Maintenance*. Butterworth-Heinemann, Linacre House Jordan Hill Oxford United Kingdom 0X2 8DP, 2 edition, 2002.

[3] FN. Verdensmålene - for bæredygtig udvikling |. https://www.verdensmaalene.dk/. (Accessed on 09/07/2022).

[4] Chris Murphy. Choosing the most suitable predictive maintenance sensor. https://www.analog.com/en/technical-articles/choosing-the-most-suitable-predictive-maintenance-sensor.html. (Accessed on 09/20/2022).

[5] 6 leading types of iot wireless technologies | behrtech blog. https://behrtech.com/blog/6-leading-types-of-iot-wireless-tech-and-their-best-use-cases/. (Accessed on 09/12/2022).

[6] ROBERT CLARK. 5g still too costly, inflexible for private network - tencent | light reading. https://www.lightreading.com/asia/5g-still-too-costly-inflexible-for-private-network---tencent/d/d-id/762170, 03 2022. (Accessed on 09/27/2022).

[7] Silicon Labs. An1137: Bluetooth mesh network performance. https://www.silabs.com/documents/public/application-notes/an1137-bluetooth-mesh-network-performance.pdf. (Accessed on 10/31/2022).

[8] Alina Bradford. What's the difference between zigbee and z-wave? https://www.safewise.com/blog/zigbee-vs-zwave-review/, 05 2022. (Accessed on 09/13/2022).

[9] Darshan Prajapati. Zigbee, z-wave, ble 5.0 and thread protocol and their applications. https://volansys.com/

selecting-right-low-power-short-range-protocols-for-your-iot-solutions/, 08 2019. (Accessed on 09/13/2022).

[10] Nordic Semiconductor. Bluetooth mesh concepts — nrf connect sdk 2.0.99 documentation. https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/ug_bt_mesh_concepts.html, 09 2022. (Accessed on 09/13/2022).

[11] A comparative study of lpwan technologies for large-scale iot deployment - sciencedirect. https://www.sciencedirect.com/science/article/pii/S2405959517302953. (Accessed on 09/12/2022).

[12] Igus. Datapro's cat7 guide and faq. https://www.datapro.net/techinfo/cat7_info.html. (Accessed on 09/23/2022).

[13] Matt Kelin. What's the difference between 2.4 and 5 ghz wi-fi. https://www.howtogeek.com/222249/whats-the-difference-between-2.4-ghz-and-5-ghz-wi-fi-and-which-should-you-use/, 07 2022. (Accessed on 09/23/2022).

[14] Kevin Walsh. Wi-fi halow™—worth the wait | wi-fi alliance. https://www.wi-fi.org/beacon/kevin-walsh/wi-fi-halow-worth-the-wait, 04 2021. (Accessed on 09/27/2022).

[15] Christian de Looper og Mark Jansen. How fast is 5g? https://www.digitaltrends.com/mobile/how-fast-is-5g/, 04 2022. (Accessed on 09/23/2022).

[16] RF Wireless World. Lte-m tutorial-features,architecture,protocol,frame,channel types. https://www.rfwireless-world.com/Tutorials/LTE-M-tutorial.html. (Accessed on 09/23/2022).

[17] RF Wireless World. What is ec gsm iot-features,applications of ec gsm iot. https://www.rfwireless-world.com/Terminology/What-is-EC-GSM-IoT.html. (Accessed on 09/23/2022).

[18] RF Wireless World. Nb iot tutorial-features,spectrum,applications of nb iot. https://www.rfwireless-world.com/Tutorials/NB-IoT-tutorial.html. (Accessed on 09/23/2022).

[19] Matter-smarthome. Faq – frequently asked questions about matter • matter-smarthome. https://matter-smarthome.de/en/matter-faq-en/#whatis, 05 22. (Accessed on 09/14/2022).

[20] Azure. What is cloud computing? a beginner's guide | microsoft azure. https://azure.microsoft.com/da-dk/resources/cloud-computing-dictionary/what-is-cloud-computing/#benefits. (Accessed on 09/22/2022).

[21] Jessica Clark. Firebase vs aws vs azure - what are the differences? https://blog.back4app.com/firebase-vs-aws-vs-azure/#Firebase_vs_AWS_vs_Azure_Comparison_Table. (Accessed on 09/26/2022).

[22] IBM Cloud Education. Iaas versus paas versus saas | ibm. https://www.ibm.com/cloud/learn/iaas-paas-saas, 09 2021. (Accessed on 09/26/2022).

[23] ProjectPro. Aws vs azure-who is the big winner in the cloud war? https://www.projectpro.io/article/aws-vs-azure-who-is-the-big-winner-in-the-cloud-war/401, 09 2022. (Accessed on 09/26/2022).

[24] Jan Steffan & Gabriel Terhorst Matthias Cäsar, Tobias Pawelke. A survey on bluetooth low energy security and privacy - sciencedirect. https://www.sciencedirect.com/science/article/pii/S1389128621005697, 03 2022. (Accessed on 09/27/2022).

[25] AWS. Aws iot security - aws iot core. https://docs.aws.amazon.com/iot/latest/developerguide/iot-security.html. (Accessed on 09/27/2022).

[26] International Society of Automation. Iec 62443. Technical report, ISA, 2007.

[27] Esp-idf programming guide. https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/esp-ble-mesh/ble-mesh-architecture.html#id1. (Accessed on 11/01/2022).

[28] Mesh Working Group. Mesh profile bluetooth specification. Technical report, Bluetooth, 2019.

[29] Mesh Working Group. Mesh model bluetooth specification. Technical report, Bluetooth, 2019.

[30] AWS. Mqtt explained - aws. https://aws.amazon.com/what-is/mqtt/. (Accessed on 12/19/2022).

[31] AWS. Aws architecture icons. https://aws.amazon.com/architecture/icons/. (Accessed on 12/20/2022).

[32] Espressif. Third-party platforms | espressif systems. https://www.espressif.com/en/ecosystem/partnership-and-resource/third-party-platforms. (Accessed on 12/30/2022).

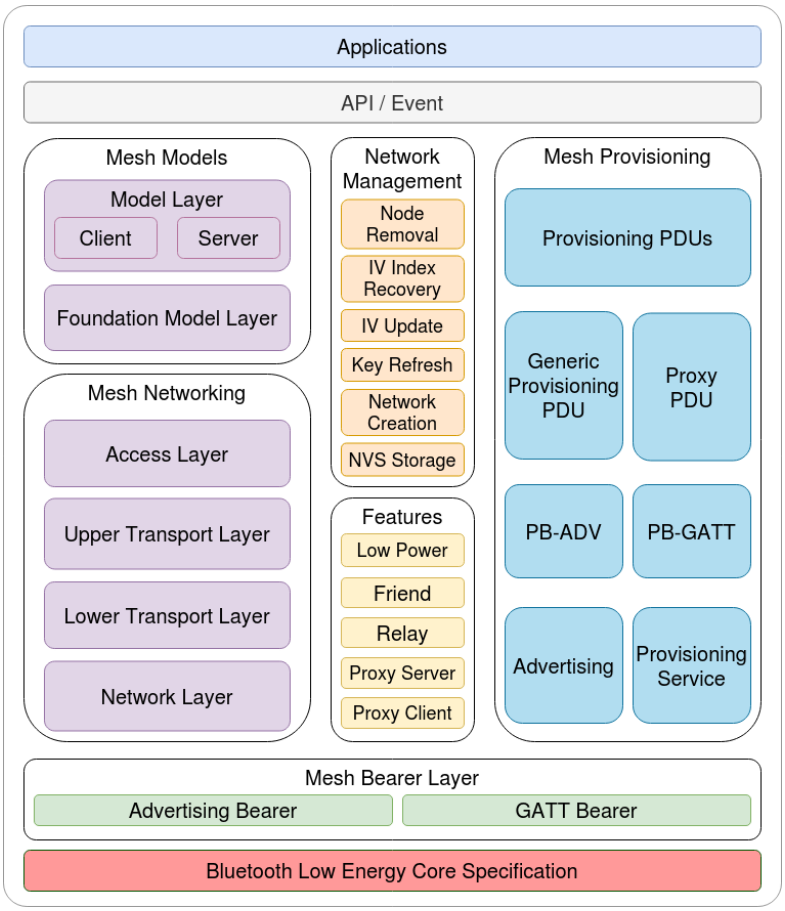
[33] Espressif. Iot development framework i espressif systems. https://www.espressif.com/en/products/sdks/esp-idf. (Accessed on 11/29/2022).

Appendiks

ESP BLE Mesh Arkitektur I

Følgende Appendiks gennemgår ESP BLE Mesh stakkens arkitektur, de forskellige grupper og deres funktionaliteter.

ESP BLE Mesh stakkens arkitektur (se på figur I.1) er opdelt i forskellige grupper, der håndterer hver sin type opgave.



Figur I.1: ESP BLE Mesh arkitektur udarbejdet af Espressif [27].

Applications implementerer applikations funktionaliteter gennem "API" kald og "Events".

API/Event håndterer "API" kald og "Events" initieret af "Applications".

Mesh Models håndterer konfigurering og integrering af "Health", "Generic", "Sensor", "Time and Scenes" og "Lighting" "Client/Server Models".

Mesh Networking håndterer kommunikationen mellem "nodes", kryptering og dekryptering af beskeder og segmentering og sammensætning af beskeder, samt ressource håndtering og "model mapping" i BLE mesh netværket.

Network Management håndterer oprettelse af BLE Mesh netværket og fjernelse af "Nodes", samt håndtering af identifikations nøgler og kommunikations index.

Mesh Features indeholder funktionaliteterne "Low Power" til energibesparelse, "Friend" til lagring af beskeder for "Low Power Nodes", "Relay" til videresendelse af beskeder og "Proxy Server/Client" til "Connection-Oriented" kommunikation.

Mesh Provisioning håndterer "Provisioning" og autentificering af "Unprovisoned" enheder, samt allokering af BLE Mesh netværk ressourcer.

Mesh Bearer håndterer dataudveksling mellem ESP BLE Mesh stakken og "Bluetooth Low Energy Core".

Bluetooth Low Energy Core er implementeringen af BLE kernefunktionaliteter [27].

Målejournaler

Elektrisk lineær aktuator harmoniske frekvenser A

Målejournals formål er at bestemme det frekvensspænd der skal kunne måles for at lave prædiktiv vedligeholdelse for den elektriske lineær aktuator. Herudfra bestemmes det tekniske krav til frekvensmåling for prototypen.

Bestemmelse af spindel rotationsfrekvens samt 1. til 4. harmoniske frekvenser for den elektriske lineær aktuator gennemgås.

A.1 Spindle rotationsfrekvens

Aktuatoren har en takometer udgang, der anvendes til at bestemme RPM under kørsel uden belastning.

Den elektriske lineær aktuator forsynes af en PSU indstillet til 25 VDC 5 A, og takometer udgangen måles med et oscilloskop.

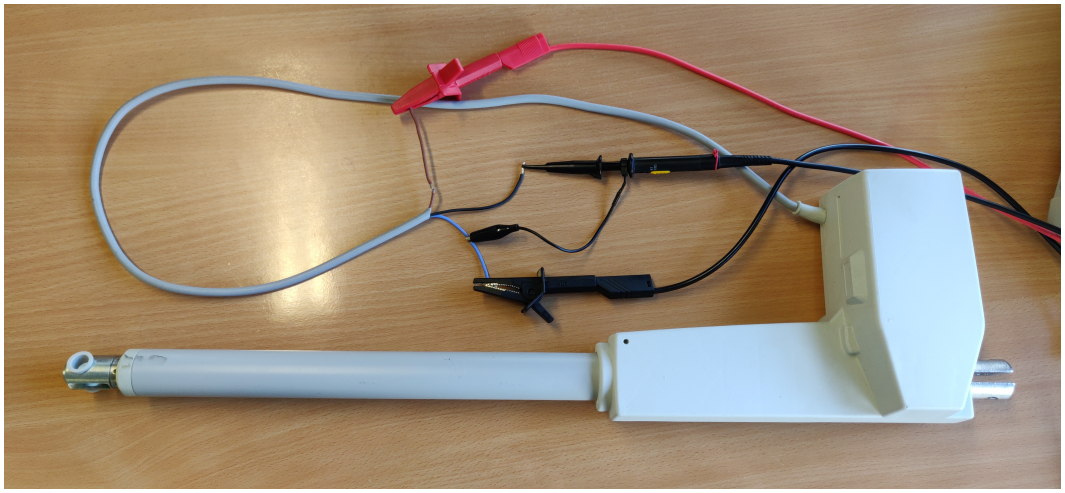
A.1.1 Materialeliste

Tabel A.1: Apparater anvendt til forsøg.

Funktion	Apparat
Elektrisk lineær aktuator	Actulux 312XXR+7036334X
PSU	Aim-TTi EX355P
Oscilloskop	Hameg HMO1024

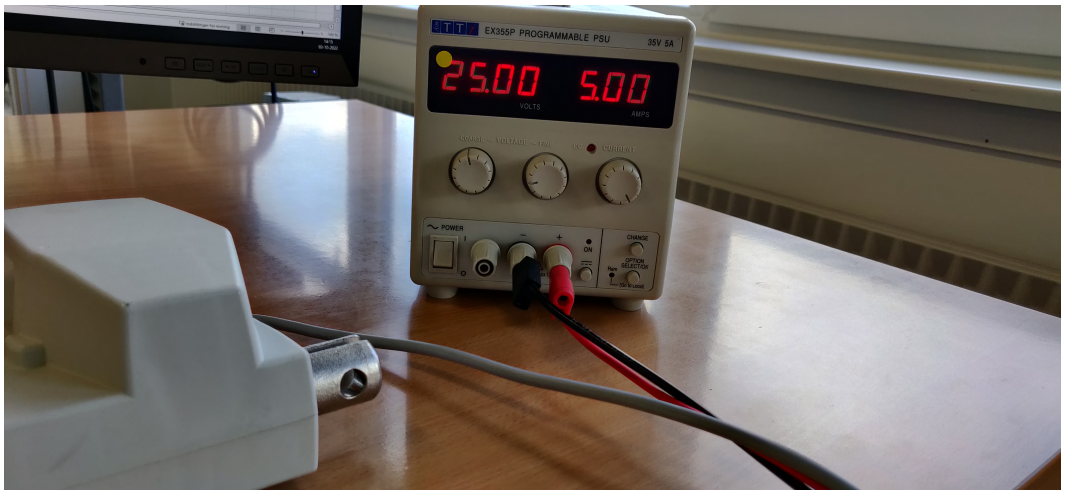
A.1.2 Opstilling

Figur A.1 viser tilslutning af positiv fra PSU (rød ledning) til aktuator forsyning 1 (brun ledning), GND fra PSU (sort ledning) til aktuator forsyning 2 (blå ledning), oscilloskop probe til takometer udgang (sort ledning) og oscilloskop GND tilslutning til GND fra PSU.



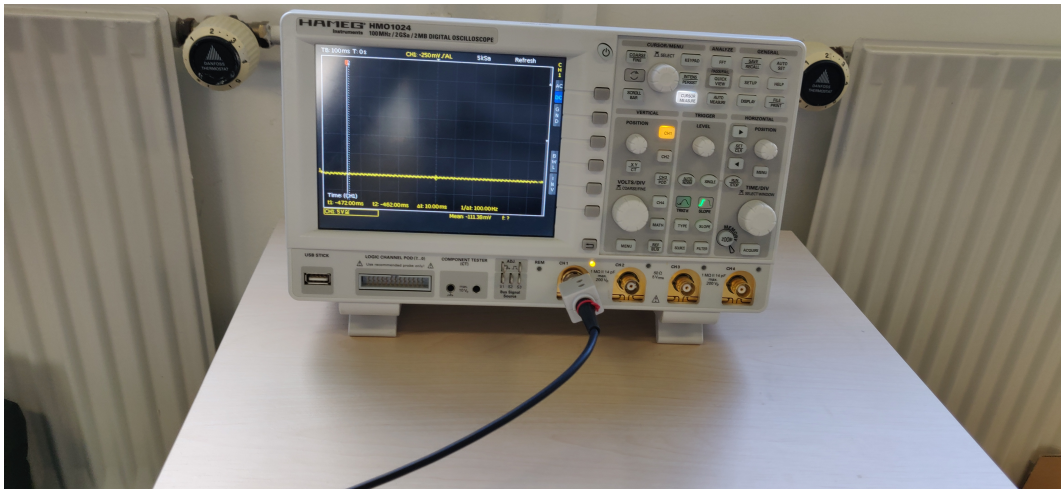
Figur A.1: Elektrisk lineær aktuator tilslutning.

Figur A.2 viser indstillingen af PSU til 25 VDC 5 A og tilslutning af positiv (rød ledning) og GND (sort ledning).



Figur A.2: PSU indstilling og tilslutning.

Figur A.3 viser oscilloskop tilslutning og opsætning.

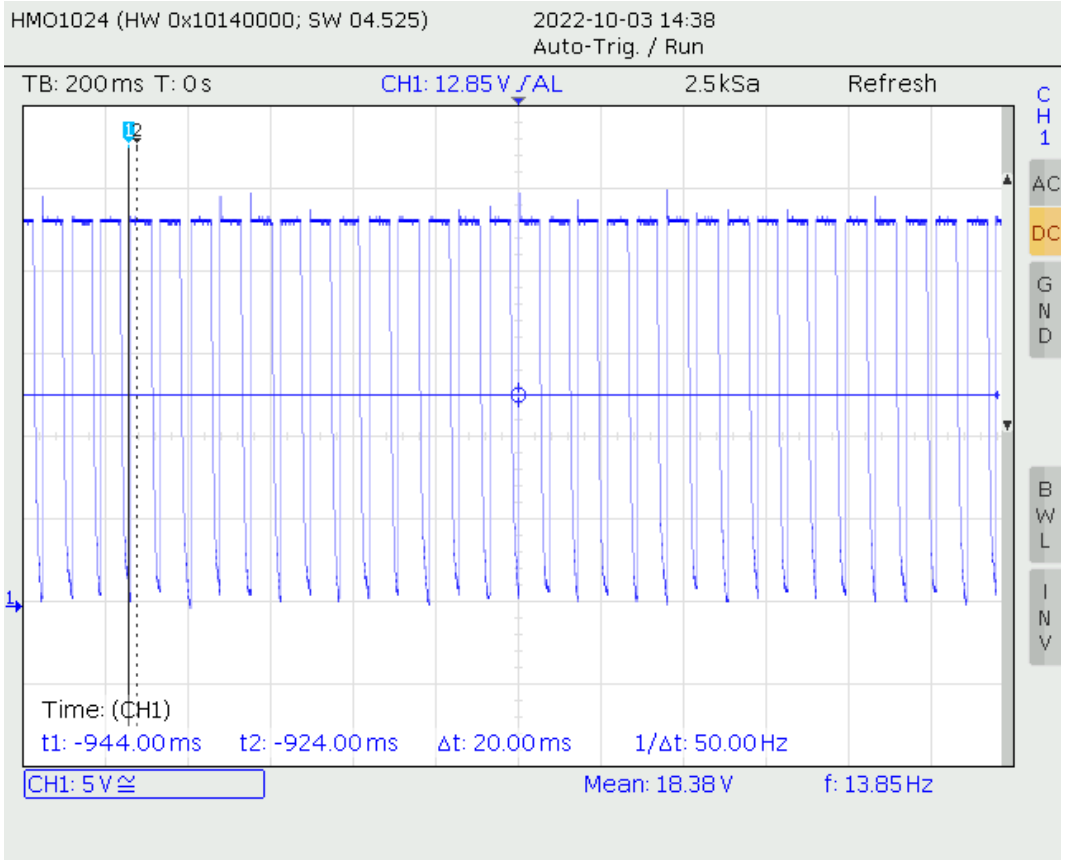


Figur A.3: Oscilloskop indstilling og opsætning.

A.1.3 Fremgangsmåde

1. Tænd PSU og indstil til 25 VDC 5 A
2. Sluk PSU output
3. Tilslut PSU plus til brun ledning fra aktuator
4. Tilslut PSU minus til blå ledning fra aktuator
5. Tænd Oscilloskop og indstil til at måle frekvens
6. Tilslut Oscilloskop probe til sort ledning fra aktuator
7. Tilslut Oscilloskop GND til blå ledning fra aktuator
8. Tænd PSU output
9. Vent til frekvensen er stabiliseret
10. Noter værdi for frekvens

A.1.4 Resultatbehandling



Figur A.4: Oscilloskop måling af frekvens.

Det kan aflæses på figur A.4 at den målte frekvens fra takometeret er 13,85 Hz. Herefter kan de harmoniske frekvenser beregnes ud fra $f_n = n \cdot f_1$ (se tabel A.2).

Tabel A.2: Harmoniske frekvenser.

Harmoniske frekvens	Frekvens
1.	13,85 Hz
2.	27,70 Hz
3.	41,55 Hz
4.	55,40 Hz

A.2 Konklusion

Ud fra tabel A.2 er den 4. harmoniske frekvens 55,40 Hz, hvilket betyder at der minimum skal kunne måles frekvenser i spændet 13,85 − 55,40 Hz for at fortage prædiktiv vedligeholdelse på den elektriske lineær aktuator.

Vibrationssensor Reaktionstid B

Målejournals formål er at bestemme reaktionstiden for Vibrationssensoren, hvorefter dette anvendes til at bestemme den tid rutinen til vibrationsmåling skal vente mellem målinger.

B.1 Reaktionstid

Udviklingsboarded anvendes til at anmode om data fra Vibrationssensoren, samt bestemme reaktionstiden.

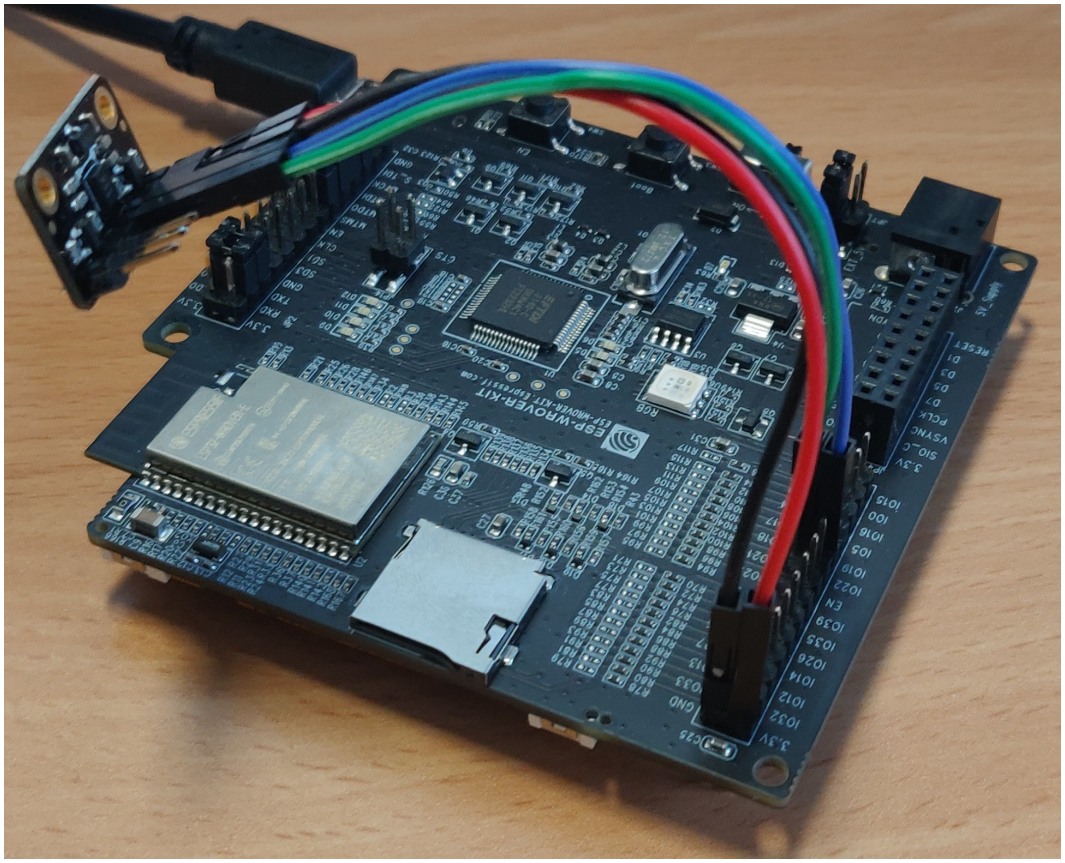
B.1.1 Materialeliste

Tabel B.1: Apparater anvendt til forsøg.

Funktion	Apparat
Udviklingsboard	ESP Wrover
Vibrationssensor	BMX160
Computer	Lenovo Thinkpad E15

B.1.2 Opstilling

Figur B.1 viser tilslutning af Vibrationssensor til ESP Wrover, hvor Vibrationssensorens VCC (Rød) til 3,3 VDC, GND (Sort) til GND, SCL (Blå) til IO19 og SDA (Grøn) til IO18. USB kabel er tilsluttet ESP Wrover og Computer.



Figur B.1: Reaktionstid opstilling.

B.1.3 Kode

For at måle reaktionstiden anvendes følgende kode, hvor tiden noteres før anmodning om vibrationsdata (**s64TimeB4**) og efter vibrationsdataene er modtaget (**s64TimeAft**). Reaktionstiden for hver måling udregnes og lægges til en variable (**u64TimeDelta**). Til sidst udregnes gennemsnittet og skrives ud på Terminal til Computer.

```
1  void BMX160_LatencyTest(void)
2  {
3      // Variable oprettes
4      struct timeval  tv_now;
5      int64_t         s64TimeB4;
6      int64_t         s64TimeAft;
7      uint64_t        u64TimeDelta = 0;
8      uint32_t        u32TimeAvg;
9
10     for(uint16_t u16X = 0; u16X < VIBRATION_DATA_LENGTH; u16X++)
11     {
12         // Tiden inden anmodning noteres
13         gettimeofday(&tv_now, NULL);
14         s64TimeB4   =   (int64_t)tv_now.tv_sec * 1000000L +
15                         (int64_t)tv_now.tv_usec;
16
17         // Anmodning om vibrationsdataene
18         tBMX160_Accel[u16X] = D_BMX160_Poll();
19
20         // Tiden efter vibrationsdataene er modtaget noteres
21         gettimeofday(&tv_now, NULL);
22         s64TimeAft  =   (int64_t)tv_now.tv_sec * 1000000L +
23                         (int64_t)tv_now.tv_usec;
24
25         // Reaktionstiden udregnes og adderes til TimeDelta
26         u64TimeDelta    += (uint64_t)((s64TimeAft - s64TimeB4));
27
28         // 8 ms delay
29         RealDelay_us(SAMPLE_DELAY_US);
30     }
31
32     // Gennemsnits reaktionstid udregnes
33     u32TimeAvg  = (uint32_t)(u64TimeDelta/VIBRATION_DATA_LENGTH);
34     ESP_LOGW(MAIN_TAG, "TimeAvg in us: %d", u32TimeAvg);
35 }
```

Kode-eksempel B.1: Reaktionstid test.

B.1.4 Fremgangsmåde

1. Tilslut Vibrationssensor til Udviklingsboard
2. Tilslut Udviklingsboard til Computer
3. Åben Terminal i VC Code
4. Noter den udregnede gennemsnits reaktionstid

B.1.5 Resultat

Den gennemsnitslige reaktionstid er udregnet til 352 μs.

B.2 Konklusion

Ud fra den gennemsnitslige reaktionstid på 352 μs og den totale ventetid på 8 ms fra ligning 5.2.2 kan ventetiden mellem målinger udregnes:

$$8\,\text{ms} - 352\,\mu\text{s} = 7{,}648\,\text{ms} \tag{B.1}$$

CbM Grænseværdier C

Målejournals formål er at bestemme grænseværdier for g-påvirkning til anvendelse i CbM udfra målinger af velfungerende elektrisk lineær aktuator.

Tabel C.1 viser 1. til 4. harmoniske frekvenser for aktuatoren, der blev bestemt i målejournal A.

Tabel C.1: Harmoniske frekvenser.

Harmoniske frekvens	Frekvens
1.	13,85 Hz
2.	27,70 Hz
3.	41,55 Hz
4.	55,40 Hz

Idet de harmoniske frekvenser er afhængige af aktuatorens RPM, der kan variere fra aktuator til aktuator, måles et frekvensspænd på ± 2 Hz omkring de harmoniske frekvenser.

C.1 CbM Grænseværdier

Prototype IIoT prædiktiv vedligeholdelsessystemet anvendes til at måle vibrationsdatasæt for den elektriske lineær aktuator, hvorefter AWS Lambda udfører FFT, samt strukturere og gemmer de beregnede data i en csv fil i AWS S3 "Bucket".

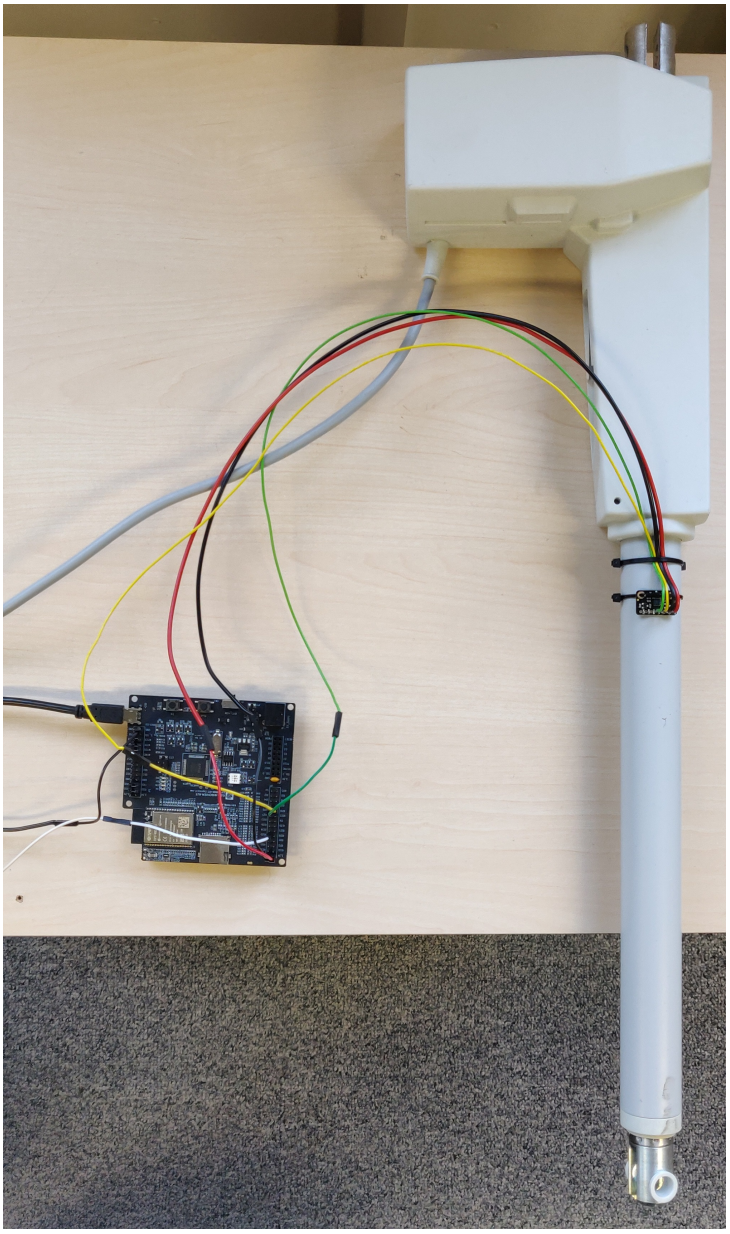
C.1.1 Materialeliste

Tabel C.2: Apparater anvendt til forsøg.

Funktion	Apparat
Elektrisk lineær aktuator	Actulux 312XXR+7036334X
Sensor Server	ESP Wrover
Vibrationssensor	BMX160
Sensor Client	ESP Wrover
PSU	Aim-TTi EX355P
Computer	Lenovo Thinkpad E15

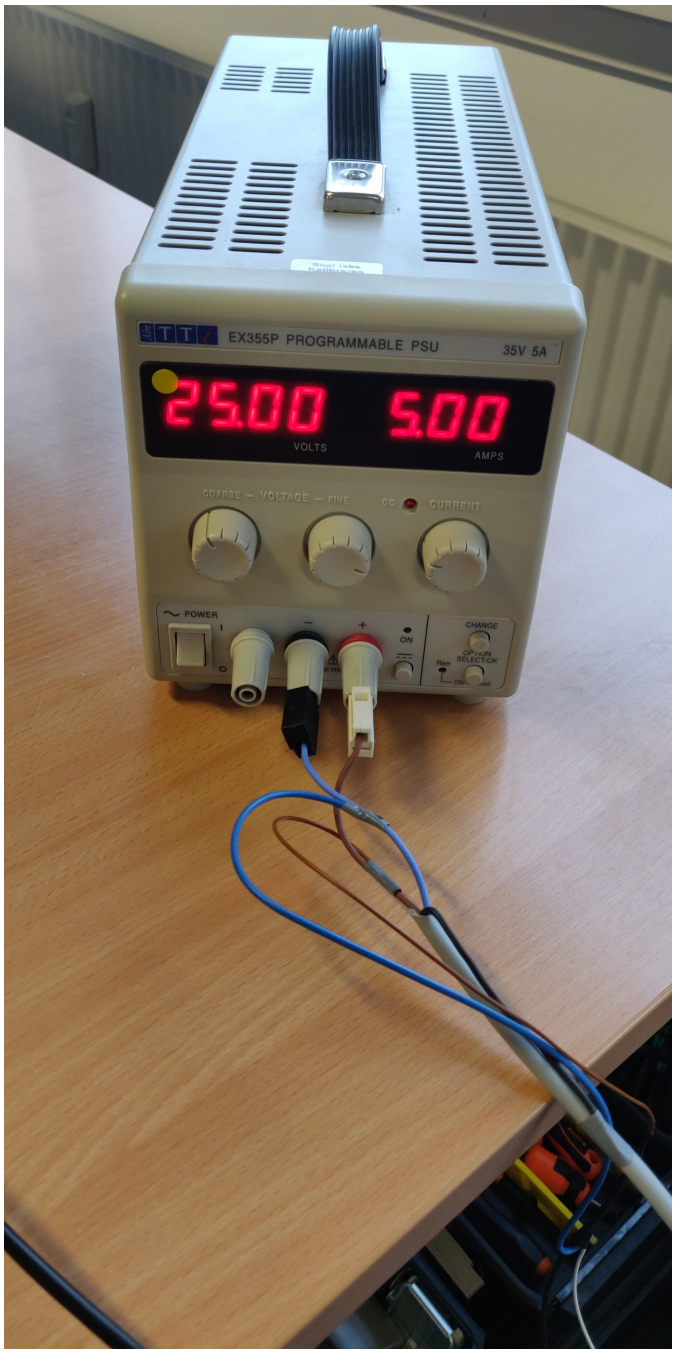
C.1.2 Opstilling

Figur C.1 viser den mekaniske tilslutning af Vibrationssensor til aktuatoren samt elektrisk tilslutning til Sensor Serveren. Sensor Server forsynes gennem USB kabel tilsluttet Computeren.



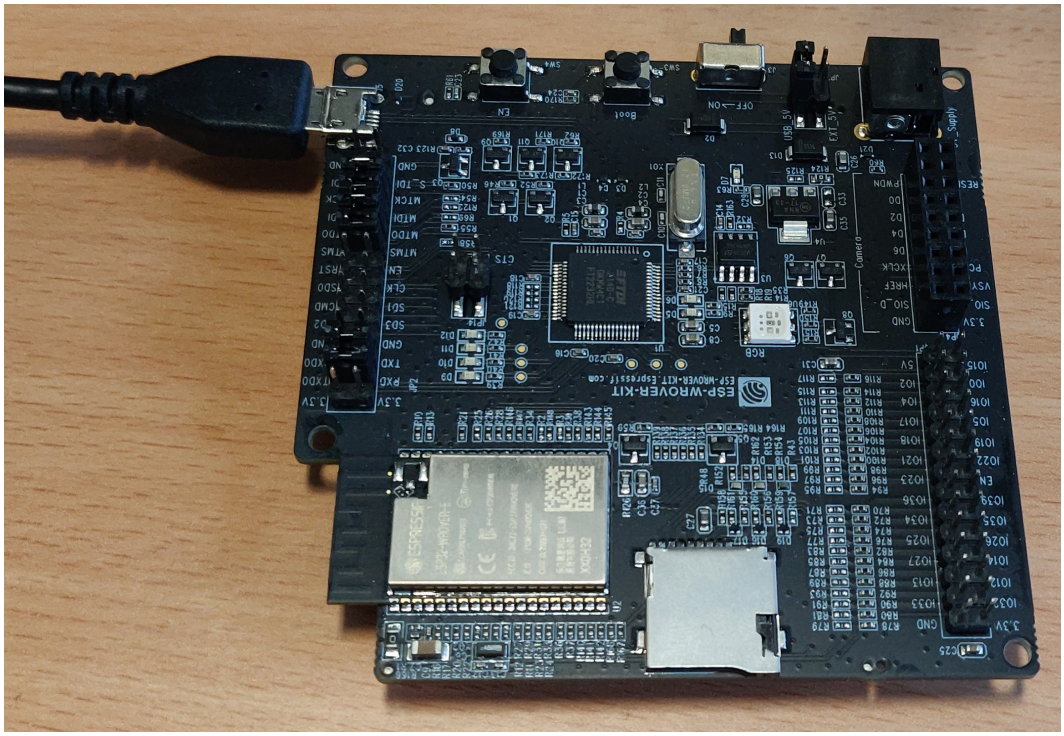
Figur C.1: CbM Grænseværdier opstilling 1.

Figur C.2 viser indstillingen af PSU til 25 VDC 5 A og tilslutning af positiv (brun ledning) og GND (blå ledning) fra den elektriske lineær aktuator.



Figur C.2: CbM Grænseværdier opstilling 2.

Figur C.3 viser Sensor Clienten forsynet gennem USB kabel tilsluttet Computeren.



Figur C.3: CbM Grænseværdier opstilling 3.

C.1.3 Fremgangsmåde

1. Tænd PSU og indstil til 25 VDC 5 A
2. Sluk PSU output
3. Tilslut PSU plus til brun ledning fra aktuator
4. Tilslut PSU minus til blå ledning fra aktuator
5. Tænd for Sensor Client og vent på RGB LED lyser grøn
6. Tænd for Sensor Server og vent på RGB LED på begge enheder bliver blå
7. Åben browser på Computer og tilgå AWS S3 "Bucket"
8. Tænd PSU output
9. Vent 3 s
10. Sluk PSU output
11. Gem den genererede csv fil med FFT data fra AWS S3 "Bucket"
12. Udlæs maksimum g-påvirkning for hver akse i frekvensspændende ±2 Hz omkring de harmoniske frekvenser
13. Gentag step 8 til 12

C.1.4 Resultat

Tabel C.3: Målinger af maksimum g-påvirkning for X-aksen i frekvensspændende ±2Hz omkring de harmoniske frekvenser.

Test	13,85 Hz	27,70 Hz	41,55 Hz	55,40 Hz
1	0,84	1,24	0,47	0,59
2	0,39	0,40	0,46	0,37
3	0,30	0,57	0,28	0,65
4	0,36	0,38	0,21	0,24
5	0,81	0,92	0,73	0,59
6	0,39	0,40	0,46	0,37
7	0,38	0,86	1,06	0,45
8	0,14	0,23	0,16	0,23
9	0,50	0,58	0,41	0,46
10	0,75	0,91	1,63	1,32

Tabel C.4: Målinger af maksimum g-påvirkning for Y-aksen i frekvensspændende ±2Hz omkring de harmoniske frekvenser.

Test	13,85 Hz	27,70 Hz	41,55 Hz	55,40 Hz
1	0,33	0,73	1,16	0,30
2	0,46	0,31	0,42	0,28
3	0,38	0,68	1,11	0,70
4	0,13	0,22	0,19	0,19
5	0,48	0,24	1,00	0,39
6	0,46	0,31	0,42	0,28
7	0,33	0,23	0,87	0,45
8	0,16	0,21	0,23	0,44
9	0,38	0,44	0,39	0,87
10	0,33	0,34	1,33	0,84

Tabel C.5: Målinger af maksimum g-påvirkning for Z-aksen i frekvensspændende ±2Hz omkring de harmoniske frekvenser.

Test	13,85 Hz	27,70 Hz	41,55 Hz	55,40 Hz
1	2,08	2,14	2,98	1,30
2	0,12	0,33	0,30	0,42
3	1,90	2,62	3,13	2,20
4	0,13	0,20	0,21	0,21
5	1,12	1,50	1,51	1,44
6	0,12	0,33	0,30	0,42
7	1,57	1,55	1,56	2,00
8	0,52	0,99	1,11	1,45
9	1,48	2,04	2,67	1,26
10	2,82	2,22	4,37	2,63

C.2 Konklusion

Ud fra de ti test i tabel C.3, C.4 og C.5 bestemmes den maksimale målte g-påvirkning for hver akse i frekvensspændende ±2 Hz omkring de fire harmoniske frekvenser se tabel C.6.

Tabel C.6: Maksimum g-påvirkning for hver akse i frekvensspændende ±2 Hz omkring de harmoniske frekvenser.

	13,85 Hz	27,70 Hz	41,55 Hz	55,40 Hz
X-akse	0,84	1,24	1,63	1,32
Y-akse	0,48	0,73	1,33	0,87
Z-akse	2,82	2,62	4,37	2,63

Accepttest af funktionelle krav D

Denne målejournal gennemgår accepttest af de funktionelle krav til prototype IIoT prædiktiv vedligeholdelsessystemet:

Tabel D.1: Oversigt over funktionelle krav.

Krav	Beskrivelse
F.1	Vibrationsmåling
F.2	Intern kommunikation BLE Mesh netværk
F.3	Ekstern kommunikation Wi-Fi internetforbindelse
F.4	Fejl Indikations Notifikation

Testen er udført med prototype IIoT prædiktiv vedligeholdelsessystemet mekanisk koblet til den elektriske lineær aktuator, hvor aktuatoren er placeret ubelastet på en plan overflade.

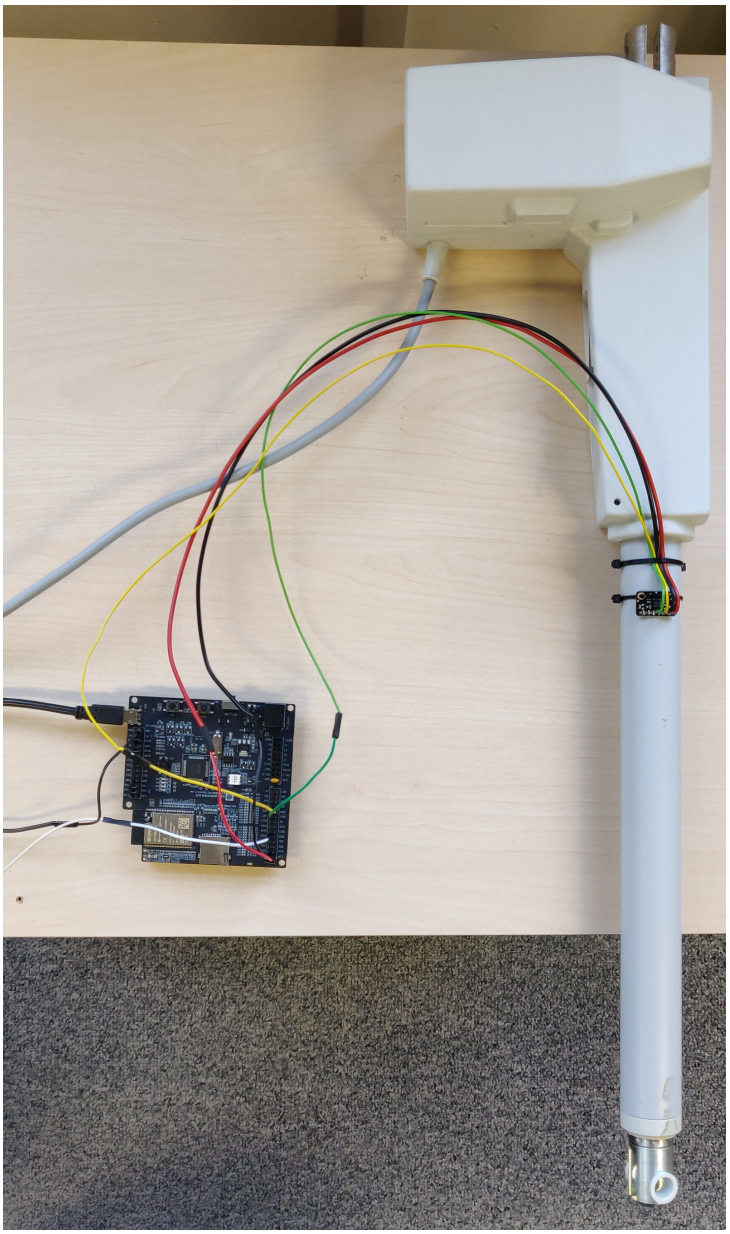
D.1 Materialeliste

Tabel D.2: Apparater anvendte til forsøg.

Funktion	Apparat
Elektrisk lineær aktuator	Actulux 312XXR+7036334X
Sensor Server	ESP Wrover
Vibrationssensor	BMX160
Sensor Client	ESP Wrover
PSU	Aim-TTi EX355P
Computer	Lenovo Thinkpad E15

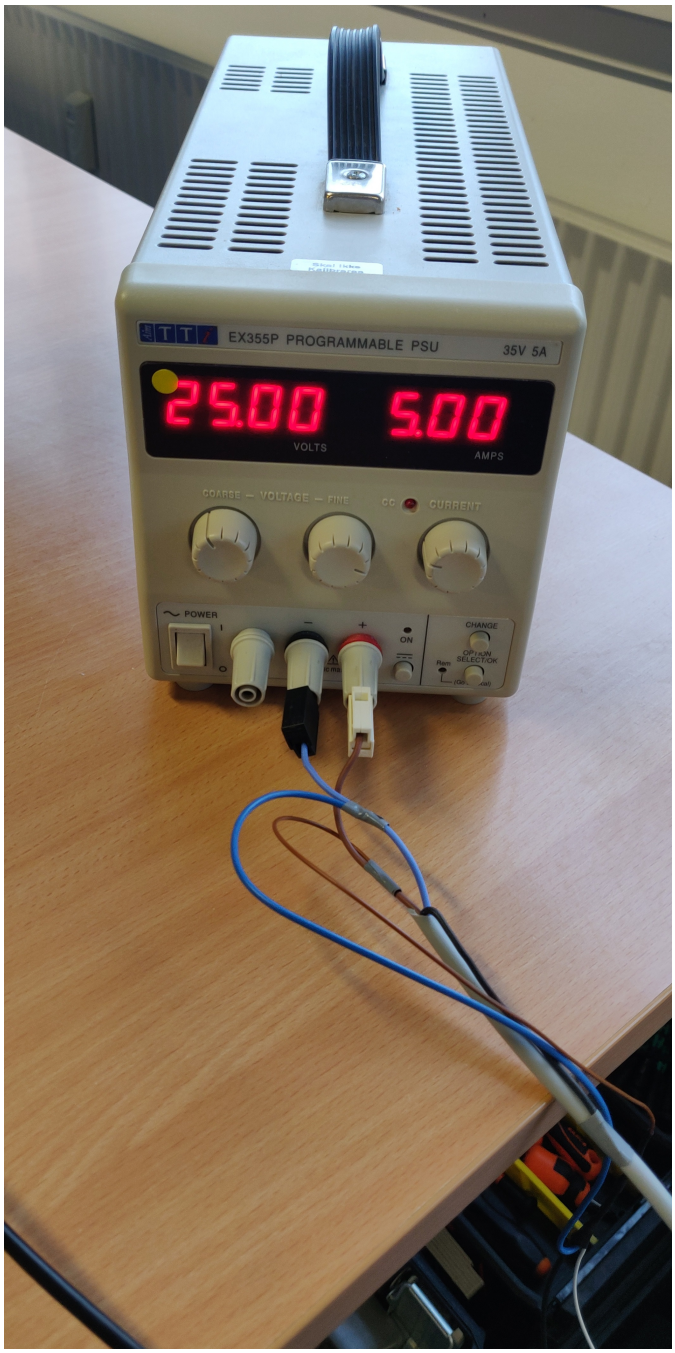
D.2 Opstilling

Figur D.1 viser den mekaniske tilslutning af Vibrationssensor til aktuatoren samt elektrisk tilslutning til Sensor Serveren. Sensor Server forsynes gennem USB kabel tilsluttet Computeren.



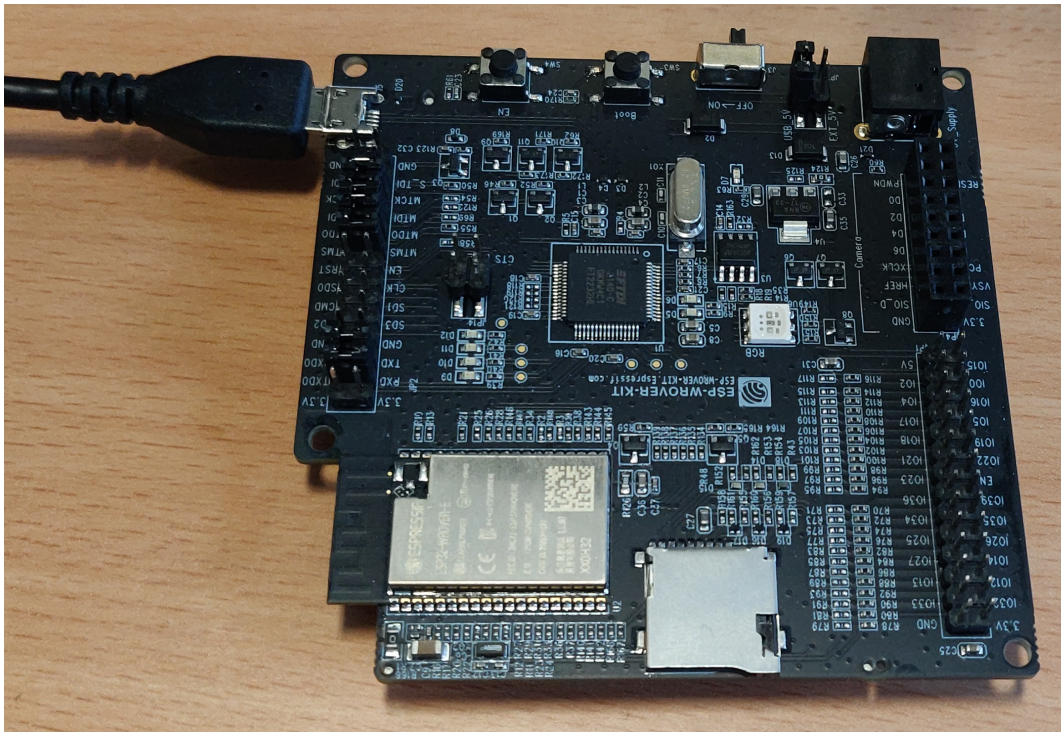
Figur D.1: Test af funktionelle krav opstilling 1.

Figur D.2 viser indstillingen af PSU til 25VDC 5A og tilslutning af positiv (brun ledning) og GND (blå ledning) fra den elektriske lineær aktuator.



Figur D.2: Test af funktionelle krav opstilling 2.

Figur D.3 viser Sensor Clienten forsynet gennem USB kabel tilsluttet Computeren.



Figur D.3: Test af funktionelle krav opstilling 3.

D.3 Fremgangsmåde

1. Tænd PSU og indstil til 25 VDC 5 A
2. Sluk PSU output
3. Tilslut PSU plus til brun ledning fra aktuator
4. Tilslut PSU minus til blå ledning fra aktuator
5. Tænd for Sensor Client og vent på RGB LED lyser grøn
6. Når RGB LED på Sensor Clienten lyser grøn har den internetforbindelse over Wi-Fi, og overholder derfor krav F.3
7. Tænd for Sensor Server og vent på RGB LED på begge enheder bliver blå
8. Når RGB LED på begge enheder lyser blå har Sensor Clienten oprettet et BLE Mesh netværk med Sensor Serveren, og overholder derfor krav F.2
9. Åben browser på Computer og tilgå AWS S3 "Bucket"
10. Tænd PSU output, så aktuatoren kører ud
11. Vent 3 s
12. Sluk PSU output, så aktuatoren stopper
13. Vent på at der er genereret en ny mappe i AWS S3 "Bucket" med tidsstempel indeholdende en csv fil med vibrations tidsdata og en med FFT data
14. Gem de genererede csv filer med FFT data og vibrations tidsdata fra AWS S3 "Bucket"
15. Valider at de generede filer indeholder data der ikke er nul, hvilket betyder at der er målt vibrationsdata, og overholder derfor krav F.1
16. Tænd PSU output, så aktuatoren kører ud
17. Påvirk den elektriske lineær aktuator i minimum 3 s mens den kører ved at ryste den under vibrationsmålingen
18. Sluk PSU output, så aktuatoren stopper
19. Vent på at der er modtaget en Fejl Indikations E-mail, hvilket betyder at krav F.4 overholders

D.4 Konklusion

I testen af de funktionelle krav kunne alle trin gennemgås, hvilket betyder at prototype IIoT prædiktiv vedligeholdelsessystemet overholder alle de funktionelle krav.

Tabel D.3: Oversigt over testede funktionelle krav.

Krav	Forklaring	✓/✗
F.1	Vibrationsmåling	✓
F.2	Intern kommunikation BLE Mesh netværk	✓
F.3	Ekstern kommunikation Wi-Fi internetforbindelse	✓
F.4	Fejl Indikations Notifikation	✓

Accepttest af tekniske krav

E

Denne målejournal gennemgår accepttest af de funktionelle krav til prototype IIoT prædiktiv vedligeholdelsessystemet:

Tabel E.1: Oversigt over tekniske krav til tests.

Krav	Beskrivelse	Værdi
T.1	Vibrationsmålings	13,85 – 55,40 Hz

Testen er udført med prototype IIoT prædiktiv vedligeholdelsessystemet mekanisk koblet til den elektriske lineær aktuator, hvor aktuatoren er placeret ubelastet på en plan overflade.

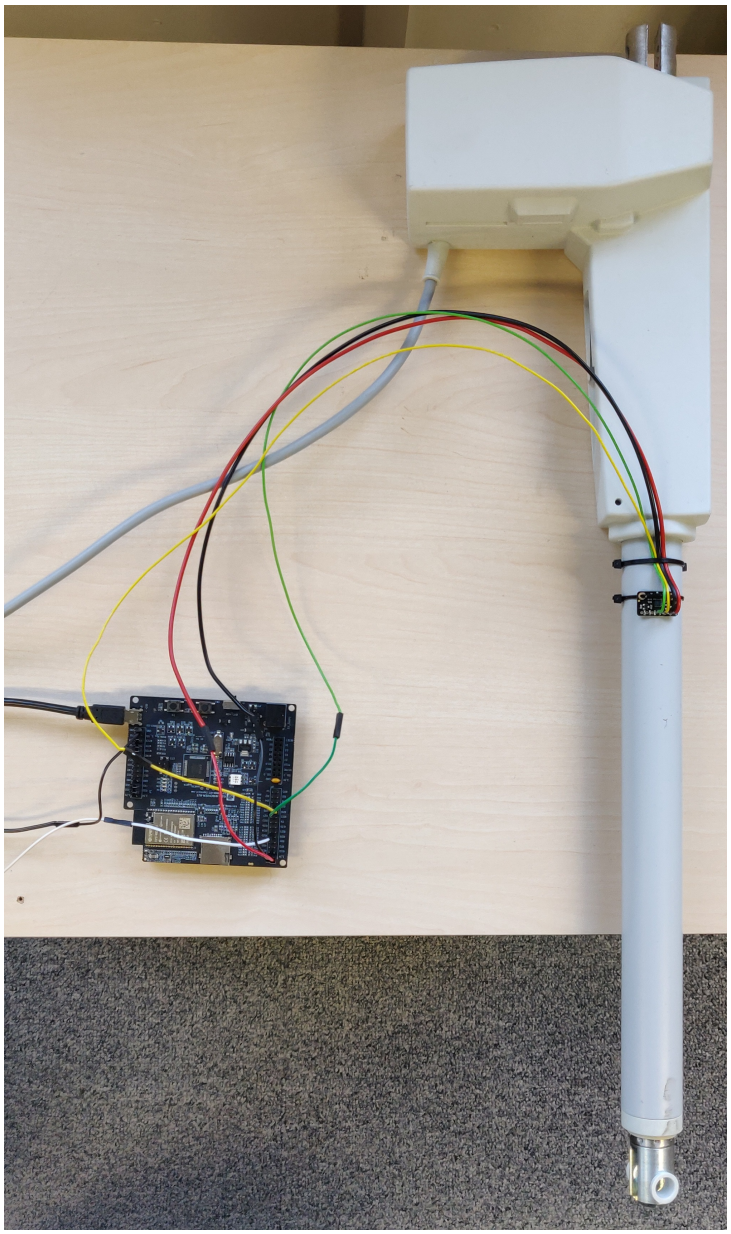
E.1 Materialeliste

Tabel E.2: Apparater anvendt til forsøg.

Funktion	Apparat
Elektrisk lineær aktuator	Actulux 312XXR+7036334X
Sensor Server	ESP Wrover
Vibrationssensor	BMX160
Sensor Client	ESP Wrover
PSU	Aim-TTi EX355P
Computer	Lenovo Thinkpad E15

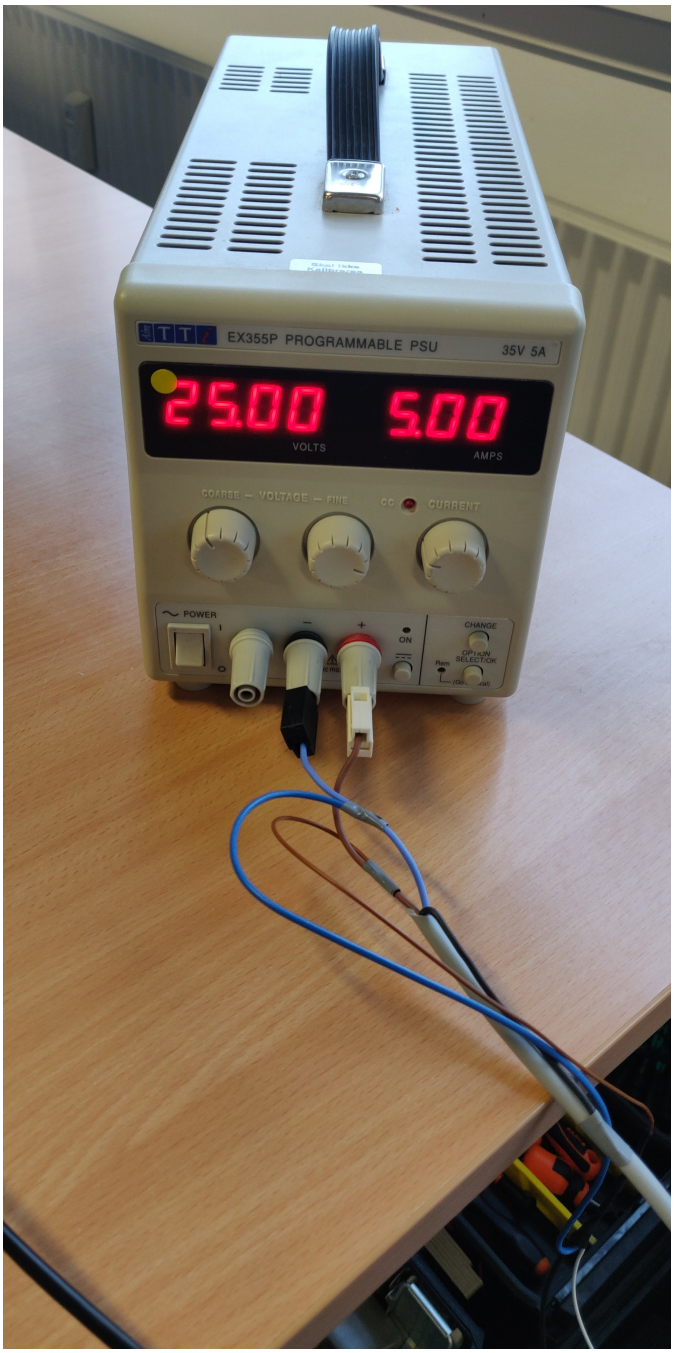
E.2 Opstilling

Figur E.1 viser den mekaniske tilslutning af Vibrationssensor til aktuatoren samt elektrisk tilslutning til Sensor Serveren. Sensor Server forsynes gennem USB kabel tilsluttet Computeren.



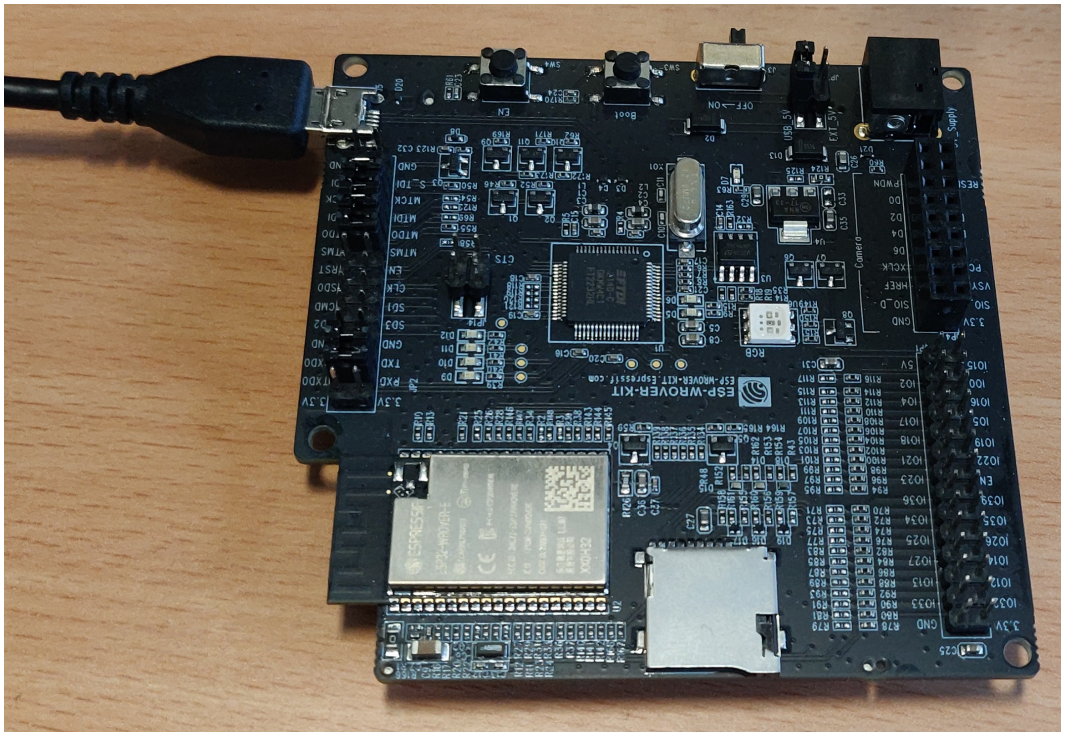
Figur E.1: Test af funktionelle krav opstilling 1.

Figur E.2 viser indstillingen af PSU til 25 VDC 5 A og tilslutning af positiv (brun ledning) og GND (blå ledning) fra den elektriske lineær aktuator.



Figur E.2: Test af funktionelle krav opstilling 2.

Figur E.3 viser Sensor Clienten forsynet gennem USB kabel tilsluttet Computeren.



Figur E.3: Test af funktionelle krav opstilling 3.

E.3 Fremgangsmåde

1. Tænd PSU og indstil til 25 VDC 5 A
2. Sluk PSU output
3. Tilslut PSU plus til brun ledning fra aktuator
4. Tilslut PSU minus til blå ledning fra aktuator
5. Tænd for Sensor Client og vent på RGB LED lyser grøn
6. Tænd for Sensor Server og vent på RGB LED på begge enheder bliver blå
7. Åben browser på Computer og tilgå AWS S3 "Bucket"
8. Tænd PSU output, så aktuatoren kører ud
9. Vent 3 s
10. Sluk PSU output, så aktuatoren stopper
11. Gem den genererede csv filer med FFT data fra AWS S3 "Bucket"

E.4 Resultater

Tabel E.3 viser første del af csv fil med FFT data fra AWS S3 "Bucket".

Tabel E.3: FFT data fra AWS S3 Bucket del 1.

Frekvens	X-akse	Y-akse	Z-akse
3,91	0,18	0,20	1,67
4,88	0,24	0,10	0,05
5,86	0,19	0,10	2,01
6,84	0,12	0,07	0,83
7,81	0,23	0,19	2,07
8,79	0,21	0,16	1,40
9,77	0,34	0,13	1,57
10,74	0,13	0,17	0,67
11,72	0,05	0,21	0,49
12,70	0,53	0,05	1,59
13,67	0,23	0,17	1,71
14,65	0,34	0,09	1,17
15,63	0,16	0,08	1,61
16,60	0,06	0,36	1,60
17,58	0,34	0,18	0,99
18,55	0,06	0,12	0,96
19,53	0,38	0,24	0,94
20,51	0,50	0,16	1,27
21,48	0,35	0,04	1,23
22,46	0,11	0,36	1,69

Tabel E.4 viser anden del af csv fil med FFT data fra AWS S3 “Bucket”.

Tabel E.4: FFT data fra AWS S3 Bucket del 2.

Frekvens	X-akse	Y-akse	Z-akse
23,44	0,45	0,47	2,10
24,41	0,81	0,72	0,34
25,39	0,52	0,49	0,89
26,37	0,53	0,24	1,22
27,34	0,16	0,31	0,76
28,32	0,29	0,23	0,31
29,30	0,12	0,24	2,51
30,27	0,35	0,43	1,14
31,25	0,10	0,44	1,35
32,23	0,26	0,38	1,19
33,20	0,56	0,52	1,53
34,18	0,35	0,31	0,81
35,16	0,39	0,58	0,81
36,13	0,19	0,16	0,58
37,11	0,14	0,19	2,34
38,09	0,39	0,07	2,13
39,06	0,23	0,27	0,28
40,04	0,44	0,20	2,00
41,02	0,06	0,26	0,63

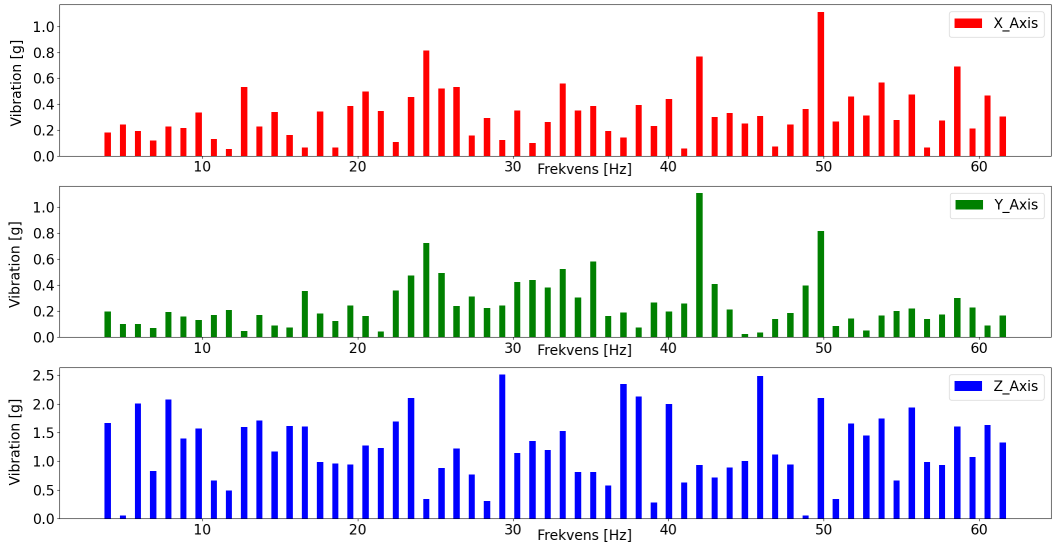
Tabel E.5 viser tredje del af csv fil med FFT data fra AWS S3 “Bucket”.

Tabel E.5: FFT data fra AWS S3 Bucket del 3.

Frekvens	X-akse	Y-akse	Z-akse
41,99	0,77	1,11	0,93
42,97	0,30	0,41	0,72
43,95	0,33	0,21	0,89
44,92	0,25	0,02	1,00
45,90	0,31	0,04	2,49
46,88	0,07	0,14	1,12
47,85	0,24	0,18	0,94
48,83	0,36	0,40	0,05
49,80	1,11	0,82	2,11
50,78	0,26	0,09	0,34
51,76	0,46	0,14	1,65
52,73	0,31	0,05	1,44
53,71	0,56	0,17	1,74
54,69	0,28	0,20	0,66
55,66	0,47	0,22	1,94
56,64	0,06	0,14	0,99
57,62	0,27	0,17	0,94
58,59	0,69	0,30	1,61
59,57	0,21	0,23	1,07
60,55	0,47	0,09	1,63
61,52	0,30	0,17	1,33

E.5 Resultatbehandling

Ud fra FFT data fra AWS S3 "Bucket" laves en graf for hver akse, hvor grafernes X-akse viser frekvens i Hz og Y-aksen g-påvirkning i g, se figur E.4.



Figur E.4: Graf med FFT data fra AWS S3 Bucket.

E.6 Konklusion

I tabel E.3 aflæses den laveste målte frekvens til 3,91 Hz, og i tabel E.4 aflæses den højeste målte frekvens til 61,52 Hz, hvilket også kan ses på figur E.4.

Det ønskede frekvensspænd fra krav T.1 er indeholdt i det målte frekvensspænd, hvilket betyder at prototype IIoT prædiktiv vedligeholdelsessystemet overholder det tekniske krav.

Tabel E.6: Oversigt over de testede tekniske krav.

Krav	Beskrivelse	Værdi	✓/✗
T.1	Vibrationsmålings	13,85 – 55,40 Hz	✓