

TEKNISK RAPPORT

Grupp 4



Dokumentinformation	
Rapportnamn	Teknisk rapport datateknik
Rapportnummer	TeknDok-<1>
Dokumentversion	0.02
Författare	Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad
Godkänd av	<Approved by>
Godkänd den	<Approval Date>

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

Sammanfattning

Projektet syftar till att utveckla ett distribuerat system för att förbättra kommunikationen och samordningen mellan brandmän i fält. Systemet består av flera noder, där varje nod representerar en brandman som kan kommunicera med närliggande noder för att dela sin position, status och uppgifter. Kraven inkluderar möjligheten att positionera noder, dela status, skapa uppgifter samt säkerställa robust peer-to-peer-kommunikation med låg latens och hög pålitlighet.

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

Innehåll

SAMMANFATTNING.....	2
INNEHÅLL	3
1 UPPGIFT.....	3
1.1 KRAV.....	4
2 PROBLEMFORMULERING	5
2.1 TEKNISKA LÖSNINGSALTERNATIV	6
3 SYSTEMÖVERSIKT.....	7
4UTREDNINGAR	7
4.1UTREDNINGAR OCH VAL AV DETALJLÖSNINGAR	18
4.2SIMULERING	22
4.2.1 Simulering av positioneringsalgoritmer.....	22
4.2.2 Simulering av den distribuerade lösningen.....	23
4.2.2.1 Scenarier	23
4.2.2.2 Simuleringsresultat	27
4.3UTREDNINGAR AV HÅRDVARA.....	27
5KONSTRUKTION AV HÅRD- OCH MJUKVARA.....	27
5.1KONSTRUKTION AV HÅRDVARA.....	27
5.2MJUKVARA	27
5.2.1Arkitekturella vyer.....	27
5.2.1.1Logisk vy	27
5.2.1.2Systemvy	30
5.2.1.3Realtidsvy	30
5.2.1.4Utvecklingsvy	30
5.2.1.5Hur olika vyer förhåller sig till varandra	31
6TESTRESULTAT FÖR SYSTEMTEST	32
7DISKUSSION OCH SLUTSATSER.....	32
REFERENSER	32

1 Uppgift

Projektet går ut på att ta fram ett distribuerat system som syftar till att underlätta koordinering mellan brandmän ute i fält. Varje nod i systemet kommer att bäras av en brandman med vilken den kommer att kunna interagera för att på ett enkelt sätt dela sin status och fråga efter hjälp av andra brandmän.

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

1.1 Krav

Funktionella Krav

KF-1.0.0.0: Positionering (**MUST**)

- Beskrivning: Varje nod skall kunna avgöra och dela sin position i med noder i närheten.
- Anledning: För att möjliggöra samordning.

KF-1.1.0.0: Dela status (**MUST**)

- Beskrivning: Varje nod skall kunna dela sin status (t.ex. "TILLGÄNGLIG", "UPPTAGEN", "BEHÖVER HJÄLP") med närliggande noder.
- Anledning: För att alla noder i närheten skall ha en uppdaterad bild av läget.

KF-1.2.0.0: Skapa uppgifter (**SHOULD**)

- Beskrivning: En nod skall kunna besluta om och tilldela uppgifter till sig själva eller andra noder baserat på deras nuvarande situation.
- Anledning: För att möjliggöra snabb uppgiftstilldelning på plats utan central kontroll.

KF-1.3.0.0: Peer-to-Peer Kommunikation (**MUST**)

- Beskrivning: Noder ska kunna kommunicera direkt med närliggande noder för att koordinera uppgifter och dela information.
- Anledning: För att möjliggöra samarbete mellan brandmän på plats utan att behöva gå via en central server.

KF-1.4.0.0: Meddelandehantering (**SHOULD**)

- Beskrivning: Noder ska kunna vidarebefordra meddelanden till andra noder så att information når fram till noder som kan vara utom räckhåll.
- Anledning: För att säkerställa att alla noder kan kommunicera med varandra.

KF-1.5.0.0: Statuskontroll för närliggande noder (**SHOULD**)

Beskrivning: Noder ska kunna upptäcka om en annan nod i närheten har slutat fungera och ta över dess uppgifter eller skicka ut en informationen till andra noder.

Anledning: För att säkerställa att systemet kan fortsätta att fungera även om en eller flera noder försvinner.

Icke-Funktionella Krav

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

IK-1.0.0.0: Robusthet (MUST)

- Beskrivning: Systemet ska kunna fortsätta att fungera även om en eller flera noder försvinner.
- Anledning: För att säkerställa systemets pålitlighet och funktionalitet under varierande förhållanden.

IK-1.1.0.0: Skalbarhet (SHOULD)

- Beskrivning: Systemet ska kunna hantera att fler än 16 noder läggs till.
- Anledning: För att kunna användas i både små och stora räddningsinsatser.

IK-1.2.0.0: Låg Strömförbrukning (SHOULD)

- Beskrivning: Noder ska kunna vara igång i minst 12 timmar på en laddning.
- Anledning: För att säkerställa att systemet kan användas under hela räddningsinsatser utan behov av laddning.

IK-1.3.0.0: Dataintegritet (MUST)

- Beskrivning: Systemet ska säkerställa att all information som delas mellan noder är korrekt och inte förloras eller förvrängs.
- Anledning: För att förhindra ofullständig information.

IK-1.4.0.0: Låg Latens (MUST)

- Beskrivning: Meddelanden mellan noder ska levereras med en maximal fördröjning på 1 sekund.
- Anledning: För att säkerställa snabb koordination i akuta situationer.

2 Problemformulering

Syftet med projektet är att designa ett distribuerat intelligent kommunikationssystem där brandmän fungerar som noder i ett nätverk. Systemet ska möjliggöra att varje brandman både kan ta emot och skicka information till andra brandmän utan att vara beroende av en central enhet eller fast infrastruktur.

Systemet måste säkerställa pålitlig kommunikation även om vissa noder, det vill säga brandmän, är otillgängliga eller befinner sig utom räckhåll. Vidare ska systemet kunna anpassa sig dynamiskt till förändringar i nätverket när brandmän förflyttar sig genom olika miljöer, såsom kollapsade byggnader eller stora brandområden. En annan viktig aspekt är att kommunikationen måste vara både säker och felfri.

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

För att uppfylla dessa krav behöver nätverket vara självorganiserande, vilket innebär att varje brandman fungerar som en autonom nod som kan kommunicera med närliggande noder. Nätverket ska också kunna självläka och omorganisera sig när noder försvinner eller nya ansluts. Realtidskommunikation är avgörande, där data som platsinformation och biometrisk data samt vital taktisk information kan utbytas direkt. Nätverket är ett ad-hoc mesh-nätverk som inte är beroende av befintlig infrastruktur och som dynamiskt anpassar sig till förändrade situationer och geografiska hinder. Slutligen måste nätverket vara robust och säkert, både mot störningar och obehörig åtkomst, samtidigt som det ska klara extrema temperaturer, fysisk påverkan och dåliga siktförhållanden.

2.1 Tekniska lösningalternativ

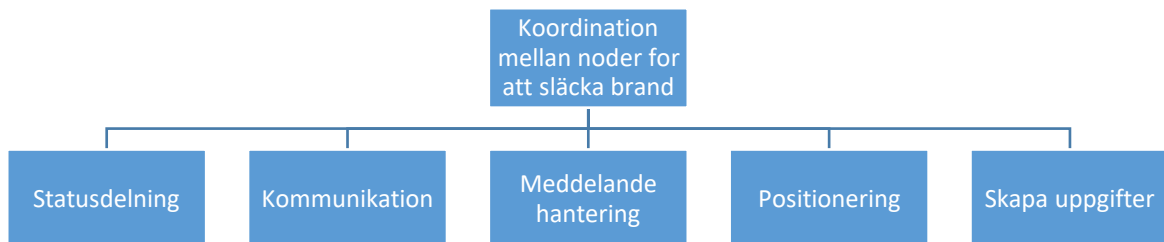
Underfunktion	Olika lösningar till underfunktionerna			
F1: Positionering	UWB	INS/IMU	PDR	
F2: Statusdelning	Fördefinierade strängar	Prioriterade integers		
F3: Kommunikation	Bluetooth	WiFi	ESP-BLE-MESH	
F4: Skapa uppgifter	Fördefinierade uppgifter	Läser in röstmeddelanden	Skriver uppgift	
F5: Meddelande.hantering	Vidarebefordra till närliggande noder	Broadcast		

Lösnings-variant	Beskrivning av lösningsvarianten
LV25	INS/IMU + Fördefinierade strängar + Bluetooth + Fördefinierade uppgifter + Vidarebefordra till närliggande noder
LV31	INS/IMU + Fördefinierade strängar + WiFi + Fördefinierade uppgifter + Vidarebefordra till närliggande noder
LV37	INS/IMU + Fördefinierade strängar + Bluetooth + Fördefinierade uppgifter + Vidarebefordra till närliggande noder
LV43	INS/IMU + Prioriterade Integers +WiFi + Fördefinierade uppgifter + Vidarebefordra till närliggande noder

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

3 Systemöversikt

Diagrammet visar en systemöversikt över ett distribuerat system för brandbekämpning, där olika funktioner samverkar för att koordinera noderna. Huvudsyftet är att möjliggöra effektiv samordning mellan noder för att hantera brandens tillstånd. Systemet delas upp i fem huvudområden: statusdelning, kommunikation, meddelandehantering, positionering och skapande av uppgifter.



- Statusdelning: Dela aktuell status mellan noder för att uppdatera om situationen eller brandens tillstånd
- Kommunikation: Säkerställa effektiv informationsöverföring mellan noderna
- Meddelandehantering: Hantera och organisera meddelanden som skickas mellan noderna, inklusive prioritering och sortering
- Positionering: Identifiera var noder befinner sig i _____ till varandra och branden
- Skapa uppgifter: Generera och tilldela uppgifter till olika noder för att släcka branden baserat på aktuella behov

4 Utredningar

För att garantera att systemet möter de tekniska kraven för noggrann positionering och pålitlig kommunikation mellan noderna har gruppen genomfört flera utredningar på olika

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

tekniker. Utredningarna inkluderar mjukvara och hårdvara analyser och har varit viktiga för att forma konstruktionsprinciperna för projektet. Tabellerna nedan visar de undersökningar som har utförts och de som gruppen valt att ta bort och inte undersöka vidare.

Genomförda Utredningar

Utredning	Syfte	Utförande	Resultat	hårdvara
Positionering: UWB	Hita exakt positionering för brandmännen	undersökt UWB kapacitet att nå hög precision när brandmän är i byggnaden	UWB valdes för den har hög precision och pålitlig	ESP 32 UWB eller UWB-sändare och mottagare.
Positionering: INS/IMU	Undersöka om INS/IMU kan användas som alternativ för positionsspårning utan externa signaler	Undersökt INS/IMU kapacitet	INS/IMU kan fungera som backup för positionering men lider av sammanlagda fel över tid som påverkar noggrannhet	IMU-sensor (accelerometer och gyroskop)
Positionering: Dead Reckoning	Undersöka om Dead Reckoning kan användas för positionering	Undersökt delkomponenterna och skapat en simulering	Hade potentiellt varit möjligt att genomföra med mer tid, men blir för komplext med tidsbegränsningen.	IMU – sensor (accelerometer och gyroskop)
Kommunikation: ESP-NOW	Avgöra om ESP-NOW kan möta kraven statusdelningen mellan brandmännen	Analys av ESP-NOW, där dess funktioner, begränsningar och styrkor granskades	Möjlig peer-to-peer lösning för direktkommunikation och statusdelning mellan noder	ESP32
Kommunikation: BLE(beacons)	Undersök om BLE med beacons kan vara en lösning för kommunikation mellan noder.	Teoretisk analys av BLE-protokollet för att undersöka dess fördelar/begränsningar	BLE har en räckvidd på 10-30 meter vilket kan vara en utmaning i större byggnader	ESP32
Kommunikation: Mesh-nätverk	Robust täckning och självlänkade nätverk	Simulering av kommunikation mellan noderna	Täckning över stora områden	ESP32

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

Kommunikation: Statuskontroll	Undersöka om Heartbeat signals kan vara en lösning	Teoretiskt analys av att använda Heartbeat signals	Effektiv metod som kontinuerligt övervakar nodernas närvaro och funktionalitet.	ESP32
Kommunikation: MOM	Stödja tillförlitlig och flexibel kommunikation mellan brandmän.	Utforskning av Meddelandeorienterad Middleware (MOM) för kommunikation mellan enheter.	Identifierade MOM:s styrkor, men beslutade att använda Mesh som primär lösning. MOM kan användas som backup vid fel.	ESP32
Tillståndsmaskin för brandmän	Optimera räddningsinsatser genom tydliga roller och förbättrad kommunikation	Utvecklade en tillståndsmaskin som beskriver arbetsflödet för grupper med specifika roller: sök, räddning och support.	Ett strukturerat system för insatser där grupper arbetar effektivt tillsammans, med flexibilitet för omplacering.	
Kommunikation: ESP-BLE-MESH	Undersöka om ESP-BLE-MESH mha heartbeats och neighbour discovery kan vara en lösning för robustheten i systemet.	Undersökt ESP-BLE-MESH, dess funktioner och begränsningar.	ESP-BLE-MESH har många passande egenskaper till projektet, som exempelvis distribuerad kommunikation, skalbarhet samt nodstatus.	ESP-32

Ultra-Wideband (UWB)

En teoretisk granskning av Ultra-Wideband UWB genomfördes för att utvärdera dess potential att erbjuda exakt positionering vilket är avgörande för att lokalisera brandmän inuti byggnader. UWB är känt för sin höga precision och tekniken analyserades för att fastställa om den kunde uppfylla projektets krav på noggrann positionering.

UWB använder korta radiosignaler med hög bandbredd för att mäta avstånd med stor noggrannhet vilket gör den särskilt lämplig för miljöer där exakt positionering är kritisk. Tekniken har den kapacitet som krävs för att leverera den precision som behövs för att möta

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

projektets mål.

En begränsning med UWB är den högre kostnaden jämfört med alternativa tekniker som INS IMU. Dessutom krävs flera sändare för att säkerställa full täckning i byggnader vilket kan göra implementeringen mer komplex. Ytterligare en utmaning är att integrera UWB med servrar för visualisering och textbearbetning vilket kan kräva extra anpassning för att synkronisera positioneringsinformationen.

INS/IMU - Utredning

Syftet med utredningen av INS/IMU var att se om denna teknik kunde leverera pålitlig positionering genom att spåra rörelsen och positionen på noderna utan att behöva externa signaler. Vårt projekt kräver en noggrann positionsuppföljning i byggnaden som brandmännen tar sig in i där GPS inte ansågs som det bästa valet bedömde vi INS/IMU som ett bra val att ha som backup eller om man vill ha låg kostnad.

INS/IMU använder en kombination av accelerometrar och gyroskop för att ständigt beräkna position och orientering vilket är bra eftersom tekniken fungerar oberoende av externa signaler vilket kan vara användbart i byggnaden när brandmännen gör sin räddning.

En nackdel som kom upp under undersökningen är att INS/IMU kan drabbas av sammanlagda fel över tid vilket betyder att noggrannheten för positioneringen kan minska ju mer systemet är aktivt utan kalibrering.

Dead Reckoning - Utredning

Inledning

Denna utredning syftar till att undersöka huruvida dead reckoning kan användas som metod för att bestämma noders position i en inomhusmiljö. Under utredningen undersöks de hårdvarukomponenter som krävs för att mäta rotation, hastighet och avstånd till en gemensam referenspunkt. Vi undersöker även dessa hårdvarukomponenters eventuella felmarginaler för

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

att fastställa om de har förmåga att låta nodernas position ha en noggrannhet på 1-2 meter.

Dead Reckoning är en metod för positionsbestämning där man beräknar ett föremåls nuvarande position genom att använda information om dess tidigare kända position samt hastighet och riktning. Denna metod bygger på kontinuerlig integration av lokal rörelsedata över tid. Genom att mäta hur ett föremål accelererar och hur det roterar kan dead reckoning-algoritmer uppskatta föremålets rörelse och därmed uppdatera dess position.

Dead reckoning används ofta i situationer där det är svårt att få tillgång till externa signaler så som GPS. För att göra detta använder systemet olika sensorer som accelerometrar och gyroskop.

ESP-NOW:

En teoretisk granskning av ESP-NOW peer-to-peer struktur genomfördes för att förstå dess möjligheter för direktkommunikation och statusdelning mellan noderna, utan behov av en central router. Varje nod kan skicka meddelanden till specifika mottagare eller via broadcast, vilket gör det möjligt för alla noder inom räckvidd att ta emot meddelanden.

Med ESP-NOW kan meddelanden riktas till specifika noder, eller sändas i broadcast-läge beroende på systemets behov. ESP-NOW erbjuder också callback-funktioner för att hantera både skickade och mottagna meddelanden, vilket gör det enkelt att övervaka meddelandens status och säkerställa att de levereras. ESP-NOW har också väldigt låg latens, vilket gör att

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

meddelanden skickas och tas emot snabbt, något som är särskilt användbart för realtidsuppdateringar i den miljön som brandmännen befinner sig i.

En eventuell begränsning med ESP-NOW är att på grund av att den är baserad på Wi-Fi-teknik, kan signalen påverkas negativt av väggar och hinder, vilket begränsar räckvidden. ESP-NOW är också bäst att använda för små datamängder, vilket innebär att det kan vara utmanande att dela större informationsmängder och detaljerad status via denna teknik. Utöver dessa begränsningar, har ESP-NOW också en begränsning när det gäller antalet enheter som kan kommunicera direkt med varandra, vilket kan påverka systemet om många enheter ska vara uppkopplade samtidigt.

BLE(beacons):

BLE stödjer både anslutningsbaserad kommunikation och broadcast-kommunikation. I detta projekt kan beacons användas för broadcast-kommunikation, där en enhet skickar ut advertising packets som kan tas emot av alla enheter inom räckvidd utan att en anslutning upprättas. Detta är fördelaktigt eftersom det minskar latens och möjliggör statusdelning med flera enheter samtidigt. Beacons kräver också minimalt med energi, eftersom de kan sända med långt sändningsintervall(minsta intervall: 20ms, största intervall: 10,24 s), vilket innebär att data kan skickas mer sällan för att spara energi. Detta gör BLE mycket energieffektivt i jämförelse med andra protokoll.

BLE har en typisk räckvidd på cirka 10-30 meter inomhus. Detta kan bli en begränsning i större byggnader eller i miljöer med många hinder. BLE räckvidd och signalstyrka kan också påverkas av material i byggnaden, som väggar eller liknande. BLE har ett relativt enkelt API för både anslutningsbaserad och broadcast-kommunikation, vilket kan underlätta implementationen. BLE-biblioteket för ESP32 tillåter justering av både advertising interval och sändningsstyrka (TX power) för att optimera energiförbrukning och räckvidd. I BLE-sammanhang handlar TX power om signalstyrkan när en enhet sänder data, vilket direkt påverkar räckvidden(Högre TX power ger längre räckvidd) och energiförbrukningen(Lägre TX power sparar batteri). En utmaning med BLE-beacons är skalbarheten i större nätverk. Om många enheter sänder samtidigt kan det leda till interferens och svårigheter att hantera data.

Mesh-nätverk:

ESP-WIFI-MESH – är ett nätverksprotokoll som klarar av ett stort antal uppkopplade noder spridda över en stor yta, såväl utomhus som inomhus. ESP-WIFI-MESH är självorganiserat och självläkande och underhålls autonomt. Till skillnad från traditionell WiFi-infrastruktur behöver inte noderna vara direkt anslutna till en central nod då de istället kan ansluta till närliggande noder. Detta ger möjlighet för ett större täckningsområde då noder som inte är inom räckvidden för den centrala noden kan istället ansluta till närliggande noder som är inom räckvidden för den centrala noden [1].

Meddelandehantering – ESP-WIFI-MESH är ett multi-hop nätverk vilket innebär att samtliga noder i nätverket är ansvarig för att vidarebefordra meddelande mellan noder, dvs de ska utöver skicka och

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

ta emot meddelande även fungera som relästation. På detta sätt kan noder skicka meddelanden till noder som är utom räckhåll från sändande nod. Detta garanterar att vilket par av två noder som helst i nätverket kommer kunna kommunicera [1].

Routing Tables- Varje nod i nätverket har en egen routing tabell för att korrekt kunna skicka eller vidarebefordra ESP-WIFI-MESH paket. Varje nods routing tabell innehåller MAC-adresser för samtliga noder inom nodens sub-nätverk. Routing tabellen används av noden för att bestämma om ett paket ska vidarebefodras uppåt- eller nedåt-ströms. För att avgöra detta används dessa regler [1]:

1. Om paketets MAC-adress finns i nodens routing tabell och inte är nodens egna, vidarebefordra paketet nedåt-ströms.
2. Om paketets MAC-adress inte finns i nodens routing tabell, vidarebefordra paketet uppåt-ströms. Om detta upprepas kommer paketet till slut hamna hos root noden vars routing tabell innehåller alla noder anslutna i mesh-nätverket.

ESP-WIFI-MESH är ett självläkande nätverk, vilket innebär att om Root Noden skulle fela kommer noderna som är anslutna till Root Noden (de som befinner sig i andra lagret) att utse en ny Root Node, där de som inte blir utsedda till Root Node bildar en uppåt-ström till denna. Om flera lager skulle fela samtidigt kommer det högst fortfarande fungerande lager att utse en ny Root Node. Skulle däremot en Parent Node fela kommer noderna som var ansluten till först att försöka ansluta till den igen. Om den misslyckas med detta efter ett antal försök kommer den att vara Idle fram till att en ny nod finns i dess räckvidd. Då bildar den en uppåt-ström anslutning till denna istället [1].

Message-Oriented Middleware (MOM)

Message-Oriented Middleware (MOM) är en infrastruktur för kommunikation och datautbyte (meddelanden). Den hanterar dataöverföring mellan applikationer genom en kommunikationskanal som bär självständiga informationsenheter (meddelanden). I en MOM-miljö skickas och tas meddelanden emot asynkront.

MOM möjliggör asynkron kommunikation, där meddelanden skickas utan att vänta på svar. Ett asynkront system använder en meddelandekö för att tillfälligt lagra meddelanden, särskilt om målapplikationen är upptagen eller inte ansluten. Klienter i en MOM-plattform kan skicka och ta emot meddelanden via denna kö, som fungerar som central för asynkron interaktion.

Middleware är en mjukvara som förbinder två eller fler objekt och förenklar komplexa distribuerade applikationer. Det innefattar webb- och applikationsservrar och är en grundläggande del i modern IT, baserat på XML, SOAP och serviceorienterad arkitektur.

- Funktioner och Kapaciteter
- Enhetlig meddelandehantering: Samlad hantering av meddelanden mellan olika

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

system och applikationer.

- Provisionering och övervakning: Tillhandahållande och kontinuerlig övervakning av resurser.
- Dynamisk skalning: Automatiserad anpassning av resurser efter behov.
- Hantera och kontrollera verktyg: Möjligheter för effektiv administration och övervakning.
- Flexibel tjänstekvalitet: Anpassningsbar prestanda för olika krav.
- Säker kommunikation: Skyddad överföring av data.
- Integration med andra verktyg: Möjlighet att integrera med andra applikationer och system.

Genom MOM kan brandmäns enheter (som sensorer och mobila enheter) utbyta information effektivt även under svåra förhållanden. MOM gör det möjligt för enheterna att publicera och prenumerera på meddelanden, vilket innebär att viktig data som plats, hälsodata eller varningar kan delas snabbt och utan central infrastruktur. Detta ökar pålitligheten och anpassningsförmågan i nätverket.

Meddelandeorienterad Middleware (MOM) kan vara användbart i detta projekt för att stödja en tillförlitlig och flexibel kommunikation mellan brandmän men vi som team beslutade att använda Mesh istället, MOM kan dock fortfarande övervägas om Mesh skulle misslyckas.

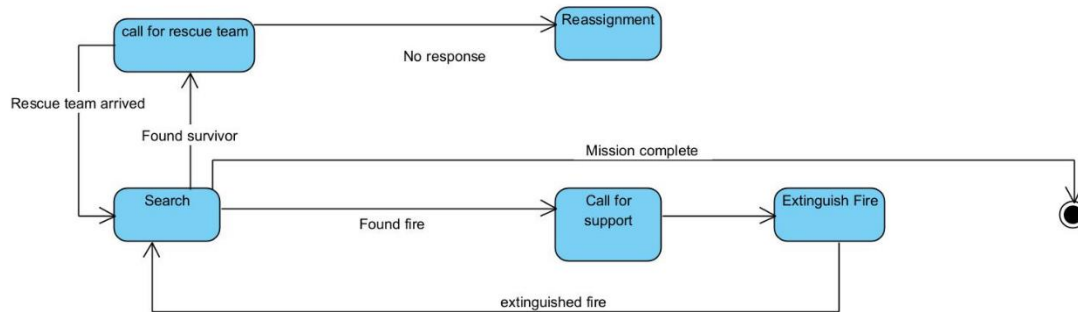
Utredning av tillståndsmaskin för brandmän

Under projektets idéstadium identifierades behovet av att optimera räddningsinsatser genom att dela upp brandmännen i specifika grupper med definierade roller. Målet var att förbättra samarbetet och kommunikationen mellan grupperna för att effektivt hantera räddningsoperationer.

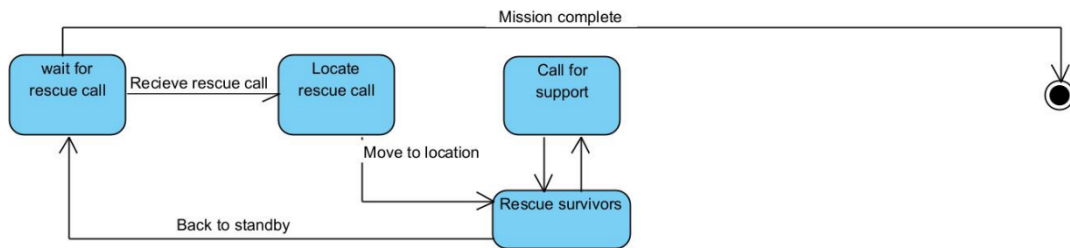
Gruppindelning och roller

Räddningsinsatsen inleds med att brandmännen delas in i tre huvudgrupper: sök-, räddnings- och supportgrupper. Varje grupp har sin egen uppsättning ansvarsområden:

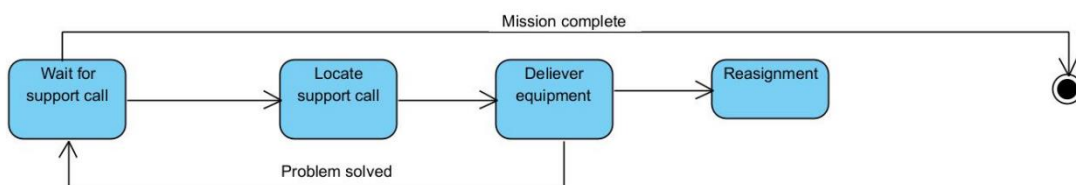
Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02



Sökgrupp: Genomför inspektioner av byggnader för att lokalisera överlevande och identifiera potentiella risker.



Räddningsgrupp: Står redo att evakueras överlevande och bistå vid räddningsinsatser.



Supportgrupp: Ansvarar för att förse de andra grupperna med nödvändig utrustning och resurser.

Tillståndsmaskinen beskriver det flöde av operationer som brandmännen utför under räddningsinsatsen. De definierade stegen i processen inkluderar:

- Divide in groups : Brandmännen organiseras i grupper utifrån deras roller.
- search: Sökgruppen går in i byggnaden för att leta efter överlevande. Om en överlevande hittas, skickas ett meddelande till räddningsgruppen.
- Call for rescue team: Om sökgruppen inte får svar kan de byta fokus till att rädda de överlevande.
- Wait for rescue call: Räddningsgruppen står i standby-läge och väntar på en begäran om räddning.
- Locate rescue call: När en räddningsbegäran tas emot, lokaliserar räddningsgruppen

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

platsen för den överlevande.

- Rescue survivors: Räddningsgruppen förflyttar sig för att evakuera den överlevande.
- Call for support: Sök- eller räddningsgruppen kan kalla på support vid behov.
- Wait for support call: Supportgruppen är i standby-läge och förbereder sig för att ge hjälp vid behov.
- Locate support call: Supportgruppen lokaliserar var stödet behövs.
- Deliver equipments : Utrustning som brandsläckare och verktyg levereras till rätt plats.
- Extinguish: Supportgruppen arbetar för att släcka branden om en sådan har upptäckts.
- Reassignment: Brandmän kan omplaceras mellan grupper beroende på situationens krav.

4.1.8

ESP-BLE-MESH

För att lösa KF-1.5.0.0 Status kontroll, kommunikation för närliggande noder samt K-1.0.0.0 Robusthet så undersöktes ESP-BLE-MESH som ett lämpligt alternativ, baserat på de breda egenskaperna samt inbyggda funktioner som medföljer.

För att uppnå robusthet och pålitlighet så behövs 3 saker.

1. Automatisk rollövertagande
2. Information till övriga noder
3. Dynamisk omdirigering

Automatisk rollövertagande

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

Ett sätt att lösa automatisk rollövertagande från nod till nod är genom att definiera och distribuera roller i förväg så har varje nod en kopia av information om närliggande noders roller med olika prioritetsnivåer som är förinställda.

Vi kan även implementera en arbetsfördelnings algoritm där alla noder är medvetna om vilka uppgifter som finns och deras prioritet.

I mesh nätverket så kan man skicka två typer av meddelanden. Broadcast, där en nod skickar ett meddelande till alla noder i nätverket, exempelvis i kritiska fall. För att inte överbelasta mesh nätverket så kan man använda sig av multicast. Detta gör det möjligt för exempelvis noder med specifika roller att skicka meddelanden till varandra.

Information till övriga noder

Med ESP-BLE-MESH så kan vi implementera “heartbeats” och “neighbor discovery” där varje nod skickar heartbeats till sina närliggande noder och övervakar heartbeats från sina grannar. Om en nod inte tar emot heartbeats från en av sina grannar så antas den vara offline. När en nod upptäcker en icke responsiv granne så kan den direkt skicka ut meddelanden till närliggande noder för att uppdatera nätverkets status.

Dynamisk omdirigering

Mesh-protokollet kan omedelbart identifiera att en nod är otillgänglig och använda en annan nod för att skicka data. ESP32 Bluetooth Mesh stödjer detta.

Följande länk visar exempel på hur man startar ESP-BLE-MESH med 3 noder.
<https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-guides/esp-ble-mesh/ble-mesh-index.html>

Utredningar som inte genomfördes

Utredning	Varför utökad genomförde inte undersöktes	Hårdvara som skulle krävas
Positionering: GPS tracking	Begränsad inomhusnoggrannhet och signalproblem ej lämplig för projektets krav	GPS-modul
Positionering: Värmesensor	Inte pålitlig säker/exakt positionering och risk för felaktiga avläsningar i byggnad. (gks: speciellt i fall av brand!)	Värmesensor
Kommunikation: Ad-hoc-nätverk	Begränsad räckvidd och kan vara mindre stabilt om noderna är för långt ifrån varandra eller om många hinder finns i vägen.	WIFI

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

< Lista/beskriv kort utredningarna som krävdes för att komma fram till konstruktionsprincipen (både-mjukvara och hårdvara), så att man får en känsla för vad som gjorts. Lista även utredningar som ni valt att INTE göra (och argumenten för det). Typiska delar är beräkningar, simuleringar, experiment, osv.>

4.1 Utredningar och val av detaljlösningar

< Förklarar vilka konstruktionsval som har gjorts och vilka alternativ som har provats. Koppla ihop tekniska lösningar med egenskaper hos produkten. Egenskaperna bör också ha en prioriteringsordning.

<x> är viktigt, så vi valde <y>, och accepterade nackdelen <z> med detta val.

Exempel: Simplicity and interoperability of the public API are priorities, so we chose REST, accepting that not all proprietary services can be implemented and be made available to 3rd party developers.

Exempel: Modifiability, scalability and cost for infrastructure are priorities, so we choose OSGi as the framework for implementing the platform, accepting this means translating device and vendor services to and from Java within the platform. .>

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

Dead Reckoning - Utredning

Inledning

Denna utredning syftar till att undersöka huruvida dead reckoning kan användas som metod för att bestämma noders position i en inomhusmiljö. Under utredningen undersöks de hårdvarukomponenter som krävs för att mäta rotation, hastighet och avstånd till en gemensam referenspunkt. Vi undersöker även dessa hårdvarukomponenters eventuella felmarginaler för att fastställa om de har förmåga att låta nodernas position ha en noggrannhet på 1-2 meter.

Vad är Dead Reckoning?

Dead Reckoning är en metod för positionsbestämning där man beräknar ett föremåls nuvarande position genom att använda information om dess tidigare kända position samt hastighet och riktning. Denna metod bygger på kontinuerlig integration av lokal rörelsedata över tid. Genom att mäta hur ett föremål accelererar och hur det roterar kan dead reckoning-algoritmer uppskatta föremålets rörelse och därmed uppdatera dess position.

Dead reckoning används ofta i situationer där det är svårt att få tillgång till externa signaler så som GPS. För att göra detta använder systemet olika sensorer som accelerometrar och gyroskop.

Hur skall Dead Reckoning-systemet fungera?

Datainsamlingen som krävs för beräkningarna skall i tanken huvudsakligen att ske via avläsning av en accelerometer och ett gyroskop. Accelerometern kommer att berätta om hur enheten accelererar och gyroskopet anger riktningen.

Accelerometern mäter förändringar i hastighet i olika riktningar. Genom att integrera accelerationsdata över tid beräknas hastigheten, och genom ytterligare integration beräknas sträckan som noden har förflyttat sig från sin ursprungliga position. Accelerometrar är dock känsliga för små fel, och dessa fel kan ackumuleras över tid.

Gyroskopet mäter rotationshastighet och används för att hålla reda på i vilken riktning noden rör sig. Genom att kombinera gyroskopdata med hastighetsdata från accelerometern kan systemet beräkna en korrekt uppskattning av både rörelse och riktning. Även gyroskop är känsliga för drift, vilket innebär att rotationsfel kan ackumuleras över tid.

Ett sätt att försöka korrigera positionen med hänsyn till de ackumulerade kommer systemet

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

att avläsa avståndet till en fast punkt. Denna referenspunkt kommer att kunna jämföras mellan de olika enheterna. Så om systemet säger att två enheter befinner sig på närliggande platser men deras avstånd till referenspunkten är väldigt olika, då har troligen något fel skett och systemet försöker då uppdatera så att positionerna stämmer. Avståndsmätningen kommer att ske med antingen WiFi eller Ultra-Wideband.

Hårdvarukomponenter och Sensorteknik

I detta avsnitt beskriver vi de olika hårdvarukomponenter och sensorer som används för att genomföra dead reckoning i en inomhusmiljö. Vi går igenom varje komponent, deras funktion, egenskaper och begränsningar för att förstå deras roll i positionsbestämningen.

- **Accelerometer:**
Accelerometern är en sensor som mäter förändringen i hastighet över tid i tre olika riktningar: x, y och z. Denna information kan användas som grund för att beräkna nodens hastighet och position genom att först integrera accelerationsvärdena (för att ta fram hastigheten) och sedan integrera den beräknade hastigheten för att få sträckan. Ett vanligt problem med accelerometrar är deras känslighet för fel, vilket kan bero på olika faktorer t.ex. vibrationsstörningar. Fel i accelerationsmätningen tenderar att ackumuleras när data integreras, vilket innebär att små avvikelser kan leda till stora fel över tid. Därför är det viktigt att undersöka om vi kan korrigera felen med någon typ av filter.
- **Gyroskop:**
Gyroskopet är en sensor som mäter rotationshastigheten runt de tre axlarna. I dead reckoning används gyroskopet för att hålla reda på vilken riktning noden rör sig i. Likt accelerometern är även gyroskop felkänsliga. Det innebär att små fel i rotationsmätningen ackumuleras över tid och leder till en avvikelse i den uppskattade riktningen.
- **Ultra-Wideband (UWB):**
UWB skulle kunna användas för att komplettera de positionsuppgifter som hämtas från accelerometern och gyroskopet. Genom att mäta avståndet till en fast referenspunkt kan detta avstånd användas för att korrigera eventuella fel som har ackumulerats över tid. UWB är speciellt användbart för noggranna avståndsmätningar tack vare dess höga precision, vilket kan hjälpa till att uppnå positionsnoggrannhet på nivå 1-2 meter.

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

Dead-Reckoning-algoritmen

Den grundläggande idén bakom dead reckoning-algoritmen är att beräkna ny position genom att följa en enkel process. Först uppskattas enhetens hastighet och riktning baserat på aktuella sensoravläsningar, och skall efter det integreras för att uppdatera positionen. Eftersom små fel tenderar att ackumuleras med tiden, innehåller algoritmen också steg för att korrigera och kalibrera positionen.

I det här avsnittet presenteras dead reckoning-algoritmen steg för steg, med fokus på hur sensordata behandlas och används för att bestämma position. Målet är att ge en översikt av hur algoritmen är tänkt att fungera och hur den kan implementeras för att bättre bestämma positionen i vårt system. Nedan följer en enkel beskrivning av algoritmen i pseudokod.

```
//Starta och ange initiala värden
```

```
Starta:
```

```
    Sätt startposition till [0, 0, 0]
```

```
    Sätt hastigheten till noll
```

```
    Sätt startriktning
```

```
    Registrera starttiden
```

```
//loop för uppdatering av position
```

```
while
```

```
    //1. Hämta data från sensorer
```

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

Hämta accelerationsvärden och rotationsvärden från sensorena

Beräkna hur lång tid som gått sedan senaste uppdatering

//2. Uppdatera riktningen

Justera riktningen baserat på rotationsvärden och den förflutna tiden

//3. Beräkna ny hastighet

Justera accelerationens effekt på hastigheten utifrån den nya riktningen

//4. Uppdatera positionen

Uppdatera positionen baserat på hastigheten och den förflutna tiden

//5. Kalibrera med referenspunkt om tillgänglig

Om en referenspunkt finns tillgänglig:

Hämta referensposition

Justera den uppskattade positionen för att matcha referenspunkten

Sätt hastigheten till noll efter kalibreringen

end

4.2 Simulering

4.2.1 Simulering av positioneringsalgoritmer

Vi har skapat en simulering först och främst tänkt att simulera positioneringen med hjälp av Dead Reckoning.

Simuleringen består av en del Java-klasser som tillsammans bygger en bild av hur en brandman skulle navigera genom en miljö, med en uppskattning av position baserad på hastighet, riktning och acceleration. Dead reckoning-algoritmen används för att uppdatera brandmannens uppskattade position baserat på tidigare position och simulerad sensorinformation.

Hur är simulationen uppbyggd?

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

Dead Reckoning och Rörelse: Brandmannens position och riktning beräknas genom en kombination av en dead reckoning-algoritm och rörelsekommandon som styrs via piltangenterna. Algoritmen använder information från en simulerad accelerometer och beräknar förändringar i hastighet och position över tid.

Sensormodellering: Simuleringen innehåller modeller av både accelerometer och gyroskop som kan tillhandahålla nödvändig data för dead reckoning-algoritmen. Accelerometern används för att beräkna förändringar i hastighet, vilket i sin tur påverkar den uppskattade positionen.

Kontroll och Interaktion: Användaren kan styra brandmannens rörelse via piltangenterna, vilket påverkar både den verkliga och uppskattade positionen.

Syftet med simuleringen

Simuleringen är tänkt att ge en uppfattning om hur dead reckoning och sensorer samverkar för att uppskatta en persons position utan yttre positioneringshjälp.

4.2.2 Simulering av den distribuerade lösningen

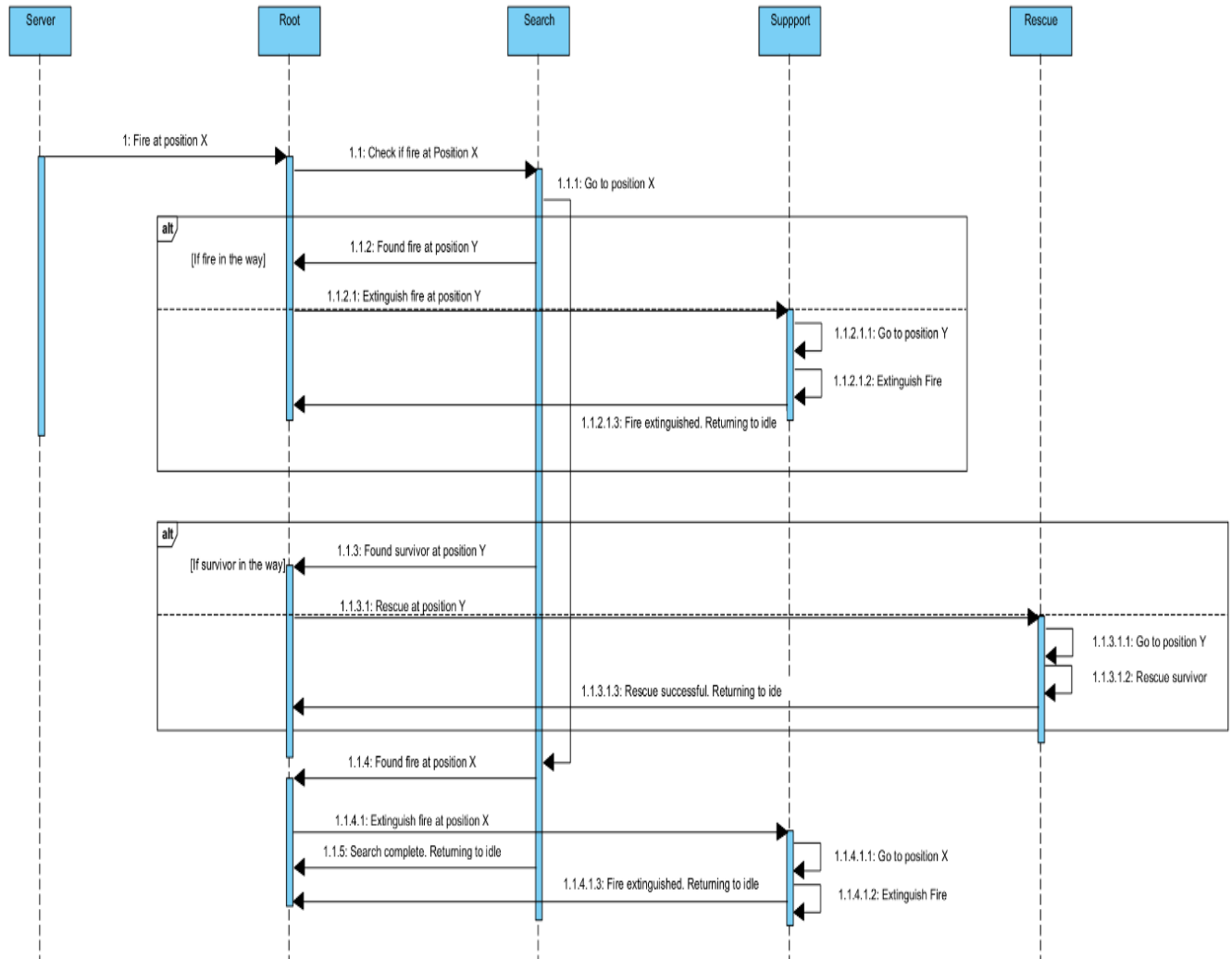
<Beskriv först hur de olika mjukvarudelar hänger ihop som behövs för att simulera det hela distribuerade systemet. Fördelen med simulering är att olika scenarier kan 'testas' och jämföras med varandra.

4.2.2.1 Scenarier

Vid brand:

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

sd [Case Fire]



Vid räddning:

Projekt:

Teknisk rapport datateknik

Projektnummer:

TeknDok-<1>

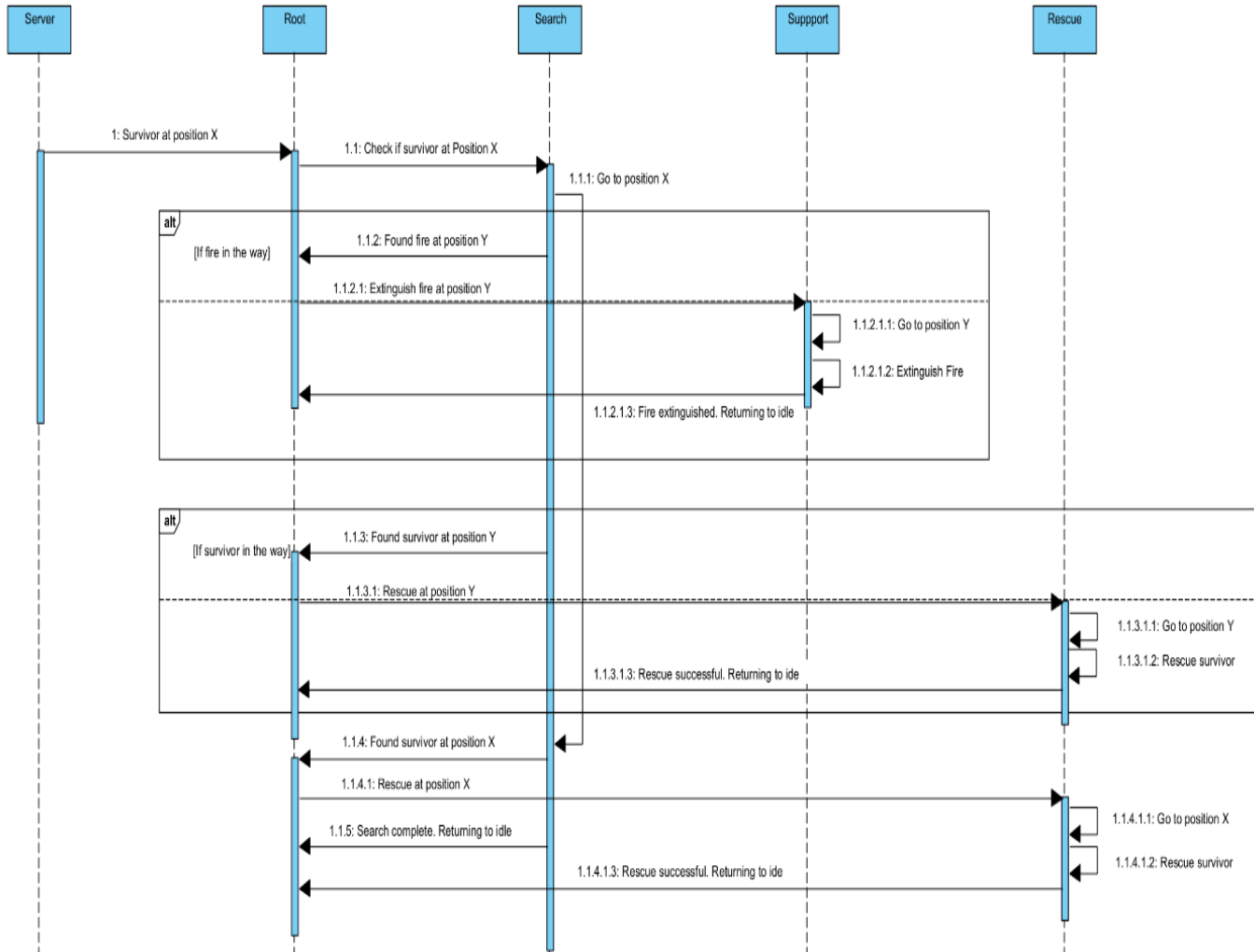
Autor:

Mohammad, Joel, Linus, Johan, Hossein, Victor,
Mohammad

Version:

0.02

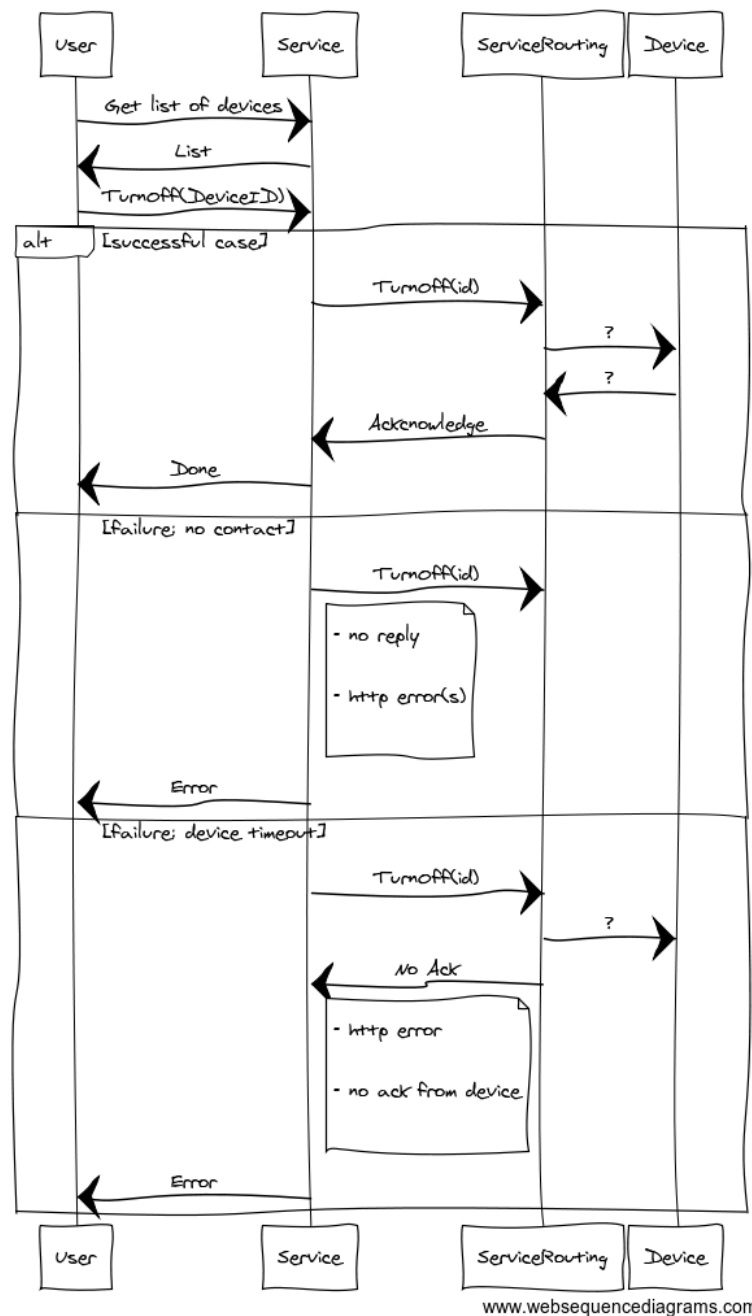
sd [Case Survivor]



< Viktiga scenarier som beskriver hur systemet är tänkt att fungera, speciellt viktigt om det finns olika delar som måste samarbeta. Scenarierna beskrivs tex med sekvensdiagram och kan även utgöra basen för simulering. Simulering kan användas för att visa/prova hur systemet fungerar i en värld av osäkerhet, tex då slumpen påverkar i vilken ordning saker händer (i ett sekvensdiagram är detta svårt att visa – det ser ofta ut som att exekveringen är ”linjär”)

Example: Physical restrictions (latency): Errors in the system are usually seen as delays or omitted responses from the devices. The following scenarios depict how the platform acts in when an error occurs when controlling a device.

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02



Figur 1: Controlling a Ninjablock device, handling timeouts and other errors. The messages marked with ? indicates that the user or platform developer has no knowledge how they look.

The error reply could contain where the error happens (how far did the request get). Inconsistencies between the user and the platform may be resolved by a device refresh.>

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

4.2.2.2 Simuleringsresultat

< Hänvisa till koden av algoritmen/algoritmerna i bilagan. Presentera resultatet från simuleringarna av helhetssystemet och jämför olika scenarier med varandra.>

4.3 Utredningar av hårdvara

< Beskriv om ni har gjort utredningar/mätningar med olika sensorer, mekaniska konstruktioner, olika material, osv.>

5 Konstruktion av hård- och mjukvara

5.1 Konstruktion av hårdvara

ESP32:

Den används som klientnoder och root Node (brandman). Root node styr kommunikation mellan klientnoderna och servern medan klientnoderna skickar positiondata och tar emot instruktioner.

Lysdiod:

användes för att indikera vilken nod som var Root node och användes som en visualliserad åtråkoppling om en händelse som brand sker lyser den.

Knapp:

knapp som användes för att signalera olika händelser som tex eld sker vill du skicka brandman till denna koordinat.

Oled display:

Displayen visar banan som ett GUI eller meddelanden som tex brandman skadad.

Kopplingar:

Alla komponenter är sammankopplade med kablar och monterade på breadboardet. För GPIO-pinnar för ESP32 och Oled Displayen har vi diskuterat och kommit överens med andra grupper om vilka pinnar som ska användas för att undvika konflikter och säkerställa korrekt konfiguration. Detta samarbete möjliggjorde en smidig integration mellan system där vi exempelvis delade information om pinnar för knappar och LED-indikatorer.

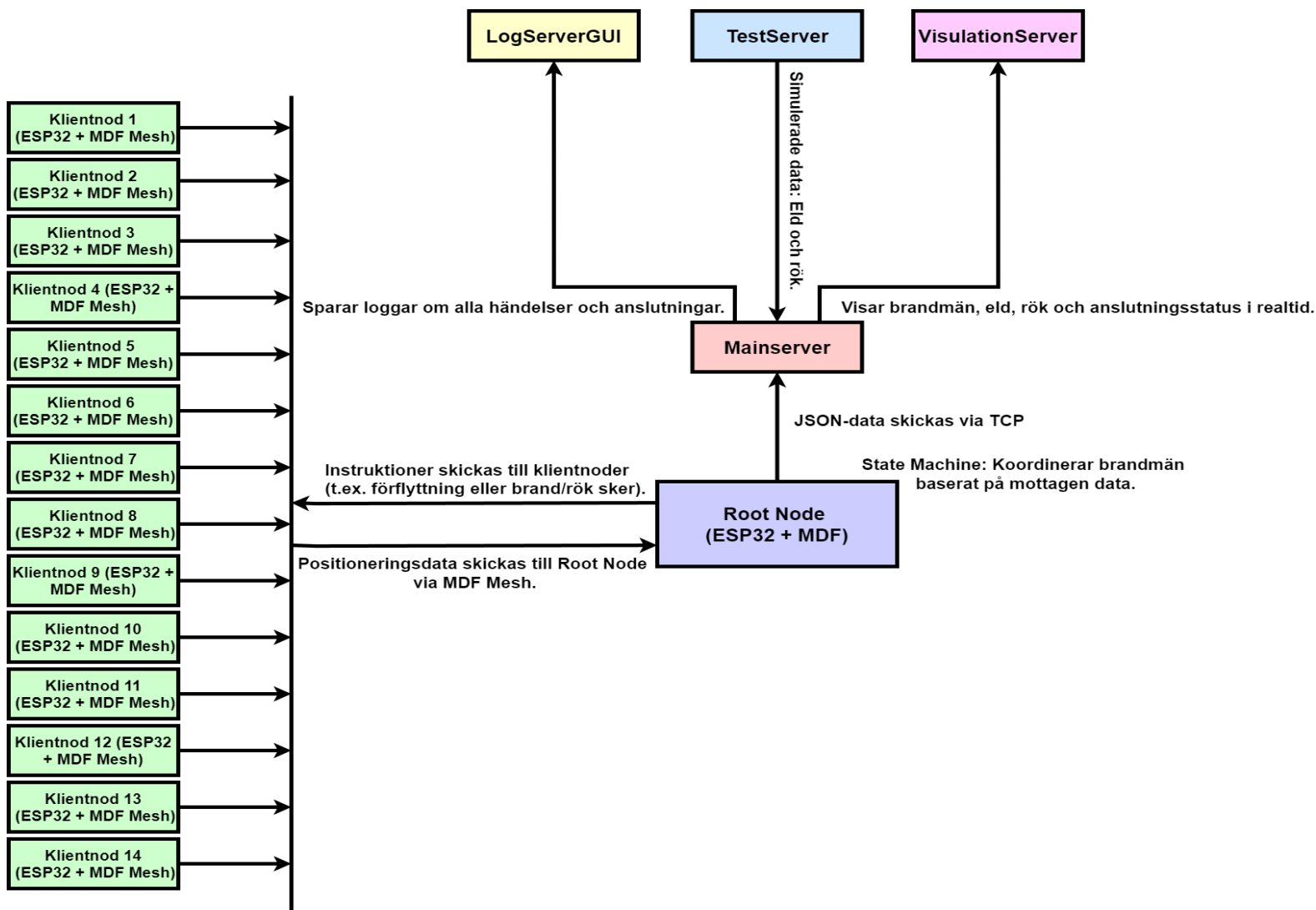
5.2 Mjukvara

5.2.1 Arkitekturella vyer

< Några få diagram som visar olika aspekter av systemet, med förklarande text så att en läsare kan förstå och tolka diagrammen. Man kan välja vilka vyer som helst som är lämpliga för sitt system, men nedanstående uppsättning vyer är väldigt vanliga..>

5.2.1.1 Logisk vy

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02



Logisk vy – beskrivning

Diagrammet ovan visar den logiska vyn för gruppens system och beskriver de olika noderna samt deras roller och kommunikationen mellan varandra.

Klientnoderna (ESP32 + MDF Mesh)

Varje klientnod (brandman) samlar in data ett exempel positionen och status som skickar dessa via MDF Mesh till Root node. Klient noderna ((ESP32) har MDF mesh koden implementerad vilket gör att den enkelt och effektivt kan kommunicera i nätverket (2.4Ghz).

Root Node (ESP32 + MDF)

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

Root node fungerar som en central node och är den som styr kommunikationen mellan klientnoderna och Mainservern. Root Noden bearbetar data från klientnoderna och kör en State maskin för att ge instruktioner till brandmännen(ESP32 noderna). Ett exempel kan vara en instruktion om förflytning, släck eld eller andra uppgifter skickas tillbaks till klientnoderna. Den skickar bearbetad data som JSON till Mainservern via TCP.

Mainserver

Mainservern hanterat all inkommande data från root Node som JSON packet och skickar vidare den data till de olika delarna av servern. Den agerar som centrum för att lagra loggar , hantera simuleringen och uppdatera visualiseringen i realtid.

LogServerGUI:

LogServer sparar loggar om alla händelser och anslutningar i systemet. Den visar som ett exempel vilka noder (brandmän) som är anslutna och deras status.

TestServer:

TestServer generar data som eld och rök. Den skickar denna data till Mainservern för att bearbeta och vidarebefordrar den data till root node, visualiseringsservern och LogServerGUI.

VisualisationServer

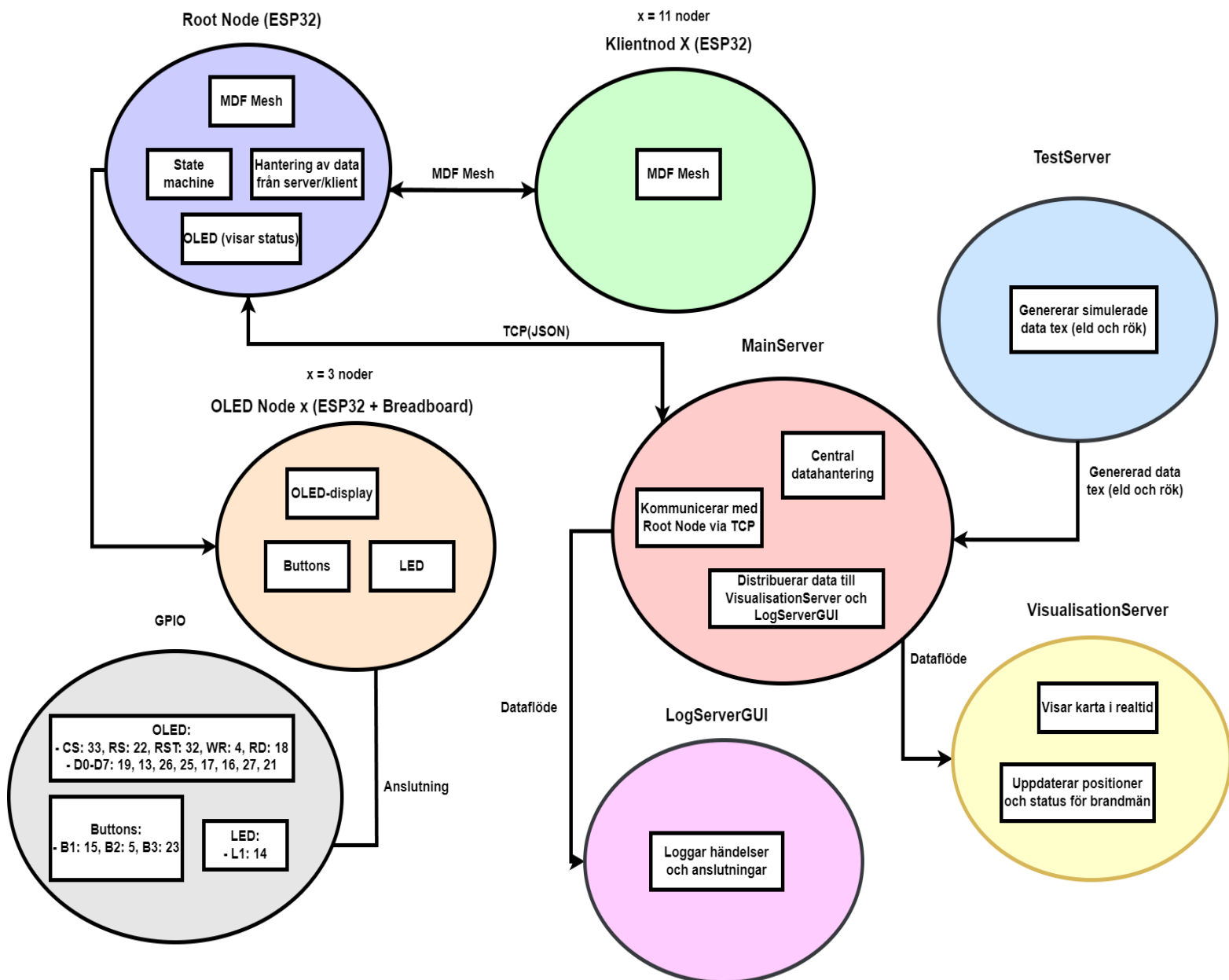
VisualisationServern visar brandmän, eld och rök på banan i realtid. Den uppdaterar kontinuerligt positionen och status för brandmännen.

Kommunikation mellan noder

1. TestServer initierar och genererar data för eld och rök som skickas till MainServer
2. MainServer skickar data som den fick från TestServern till Visualiseringsservern, LogServerGUI och Root Node. Tex Eld och rök vissas på visualationsservern och LogServern Loggar händelserna.
3. Root node får data från MainServern och skickar instruktionerna till Klientnoderna via MDF Mesh.
4. Klientnoderna skickar bearbetade data till MainServern som distribuerar den till LogServerGUI och visualationsservern uppdaterar banan i realtid.

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

5.2.1.2 Systemvy



Fysisk Vy beskrivning

Diagrammet ovan visar den fysiska arkitekturen för projektet. Det visar hur olika noder och komponenter är kopplade och vilken hårdvara som används.

Root Node(ESP32):

Hanterar kommunikationen med klientnoder via MDF Mesh

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

Klientnoder (ESP32):

tar emot instruktioner från Root node via MDF Mesh. Totalt används 14 klientnoder

Oled Node (ESP32 + Breadboard) :

Består av 1 ESP32 kopplad till en Oled display , knappar och led via gpio. Oled visar meddelanden , knapparna används som bekräftelse tex (hjälp behövs , ja eller nej) och led indikerar eld. Den får meddelanden från Root node.

Server:

MainServer:

Centraliserad datahantering och kommunicerar med Root Node via TCP(JSON). Distribuerar information till LofServerGUI och VisualServer.

TestServer:

Genererar simulerad data tex (eld och rök) och skickar till MainServer.

VisualisationServer:

Visar banan i realtid och uppdaterar positioner och status för brandmän.

LogServerGUI:

Loggar alla händelser och anslutningar.

5.2.1.5 Hur olika vyer förhåller sig till varandra

Den logiska vyn beskriver systemets arkitektur på en hög nivå inkl noder och deras roll i kommunikationen. Den fysiska vyn kompletterar detta genom att visa den faktiska hårdvaran och kopplingarna mellan komponenter. Ett exempel vilken mikroprocessor (ESP32) som kör vilken kod och hur GPIO används för att styra OLED-display, knappar och LED/lysdiod.

Båda vyerna samverkar genom att den fysiska vyn tydliggör hur den logiska vyn implementeras på hårdvarunivå. Root Node som beskrivs i båda vyerna är ett bra exempel på detta där dess funktioner i

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

den logiska vyn (tex kommunikation via MDF Mesh och hantering av data) är knutna till specifik ESP32 hårdvara i den fysiska vyn.

6 Testresultat för systemtest

<Sammanställning av testresultat från system-tester som beskrivs i testdokumentet. Illustrera med bilder och eventuell video (länkar till egen Youtube-sida). Sammanfatta testerna här: Vilka krav (från listan i avsnitt 1.1 ovan) uppfyller produkten inte och varför inte?

Exempel av testrapport:

Test case/ version tested	Version 0.7 (2014-05-15)	Version 0.9 (2014-05-26)
TC1	Pass (u-tube-link)	Pass
TC2	Fail (link...)	Pass (u-tube-link)
...		

7 Diskussion och slutsatser

Projektet har hittills resulterat i en fungerande integration av flera komponenter som tex kommunikation mellan Root Node, klientnoder och servrar men en del av system som tex Oled displayen och brandman logiken är inte ännu helt färdigställd.

Under arbetet har vi identifierat och löst flera utmaningar som tex kommunikation mellan servrar och noder samt GPIO konfiguration för hårdvarukomponenter.

Trots framsteg är det tydligt att projektet inte kommer bli färdig inom den ursprungliga tidsramen. Erfarenheterna från detta projekt har dock varit värdefulla och bidragit till ökad förståelse för utveckling av inbyggda system och distribuerad kommunikation.

Framöver planerar vi att slutföra OLED displayen implementation och slutliga logiken för att slutföra projektet. De återstående delar inklusive Oled display och logiken planeras att färdigställas efter

Projekt: Teknisk rapport datateknik	Projektnummer: TeknDok-<1>
Autor: Mohammad, Joel, Linus, Johan, Hossein, Victor, Mohammad	Version: 0.02

deadline.

1. Espressif Systems. (2023). ESP-BLE-Mesh User Guide.
2. PCforAlla. (2023). *Ultra Wideband (UWB) – så fungerar den trådlösa tekniken.*
3. <https://www.pcfforall.se>
4. PCforAlla. (2023). *Vad du behöver veta om UWB positioneringsteknik med 10 cm noggrannhet.*
5. <https://www.pcfforall.se>
6. Inertial Sense. (2023). *Difference Between IMU, AHRS, & INS - Overview* [Video].
7. <https://www.inertialsense.com>
8. JOUAV. (2023). *A Complete Guide to Inertial Measurement Unit (IMU).*
9. <https://www.jouav.com>
10. Espressif Systems. (2023). *Bluetooth GAP API Reference.*
11. https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/bluetooth/esp_gap_bt.html
12. Espressif Systems. (2023). *Getting Started with BLE Device Discovery.*
13. <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/ble/get-started/ble-device-discovery.html>
14. Espressif Systems. (2023). *ESP-NOW API Reference.*
15. https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/network/esp_now.html
16. Espressif Systems. (2023). *Modifying TX Power for Advanced Development.*
17. <https://docs.espressif.com/projects/esp-techpedia/en/latest/esp-friends/advanced-development/performance/modify-tx-power.html>