

Projektplan: Kartläggning av Miljö

Sebastian Ling, Gustav Löfqvist & Zimon Moudi

Möjligheten att kartlägga övergivna städer oberoende av deras karaktär har stor tillämpning inom flera användningsområden. Det är ett uppdrag som är särskilt lockande för autonoma robotar. De kan justeras för att kunna kartlägga flexibelt och anpassningsbart. Vårt arbete innebär att utforska de olika tekniska möjligheterna genom att undvika traditionell hårdkodning. Vi kommer istället försöka tillämpa smarta och nyskapande tekniska lösningar.

Denna rapport är en guide genom vårt projekt, från dess grundläggande syfte till de mer avancerande och tekniska detaljerna som utgör dess ryggrad.

Problemformulering

Vårt uppdrag är att utveckla en autonom robot som kan kartlägga varierande miljöer oberoende av deras karaktär. Hur ska en autonom robot ha kapaciteten för att kartlägga oberoende av miljö? Det är en mycket komplicerad utmaning som kräver påhittiga lösningar, pålitlig hårdvara, smidig manövrering och sofistikerad mjukvara som kan tolka och anpassa sig till den omgivande världen.

Vårt mål är att utveckla hård- och mjukvara som möjliggör en sådan automation. För att lyckas behöver vi övervinna flera utmaningar:

- **Pålitlig hårdvara**
Roboten måste vara utrustad med robust hårdvara som inkluderar men inte begränsas till: sensorer, kameror och fysisk plattform som kan hantera styrning och acceleration.
- **Mjukvara som tar smarta beslut**
Den nödvändiga mjukvaran måste effektivt behandla information från sensorer och kameror. Den måste fatta vissa intelligenta beslut och göra slutsatser utifrån den data som är tillhands. Och därefter skicka exakta styrkommandon till robotens rörelsekomponenter.
- **Anpassningsbarhet och flexibilitet**
Mjukvaran behöver ha förmåga att anpassa sig efter olika miljöer och omständigheter. Den ska kunna hantera föränderliga förhållanden för att möjliggöra flexibel kartläggning.
- **Kommunikation med central enhet**
För att tillåta någon form av fjärrövervakning och –kontroll, behöver roboten ha en öppen kommunikationskanal.
- **Inget rum för ineffektivitet**
Eftersom robotens system är sammanlänkade, kan ineffektivitet i en del av något system tillföra stor påverkan på helheten. Detta kräver att alla komponenter fungerar smidigt och effektivt.

En autonom kartläggande robot har en mängd potentiella användningsområden, inklusive utforskning av okända eller farliga miljöer, inspektion av infrastruktur och exempelvis stöd i räddningsarbeten. Det är en värdefull resurs i många situationer där mänsklig närvaro är riskabel.

Hårdvarusystem

Vår vision är att robotens hårdvarukonfiguration ska vara en noggrant utvald ansamling komponenter som kan arbeta i harmoni. Komponenterna ska hantera kartläggning av varierande miljöer. Hjärnan av roboten kommer bestå av en MaixPy M1S, den skall fungera som den intelligenta kärnan som tar hand om datorvision- och AI uppgifter. Den har kapacitet att tolka bilder i realtid vilket är väsentligt för att klara av att justera sig efter omgivningen och navigera på ett säkert samt effektivt sätt (följa väg).

Som komplement till MaixPy-enheten ska vi inkludera en pålitlig arduino-mikrokontroller, arbetshästen som kommer sköta motorstyrning och rörelsestyrning. Mikrokontrollen avlastar vår MaixPy samtidigt som den samordnar våra sensorer och samlar värdefull data om omgivningen.

En av de mest kritiska komponenter är kameran, vilket kommer fungera som robotens öga. Kameran tar bilder och möjliggör för bildigenkänning och visuell uppfattning. Kameran har kapaciteten att identifiera markeringar och mönster som används för att vägleda roboten. Ett tillskott till kameran är exempelvis avståndssensorer, gyroskop, accelerometer och IR-sensorer. Det är sensorer som kan förenkla uppgiften och förebygga problem. Visionen är att utveckla en mindre och smidig robot. Därför kommer vi endast implementera de nödvändiga sensorerna och funktioner. På grund av det kommer vi noggrant diskutera varje komponent och klargöra dess betydelse.



Vi har i tidigare projekt använt oss av olika variationer till motorn på bilden. Efter granskning har vi beslutat att motorn troligtvis inte är applicerbar i vårt projekt, på grund av dess storlek. Vi letar efter mindre alternativ som inte kompromissar på prestandan. Ett förslag är att vi återanvänder motorn och utvecklar en egen växellåda alternativt finner en mer passande modell.

Tillsammans kommer komponenterna skapa en mångsidig och robust hårdvarukonfiguration som blir avgörande för vårt projekts resultat.

- **MaixPy SiSpeed M1 Dock**

MaixPy SiSpeed M1S är kärnan i vårt projekt och fungerar som den centrala intelligensenheten. Den är baserad på en SiSpeed M1-chip med dedikerad AI- och datorvisionskapacitet.

MaixPy-enheten hanterar bildigenkänning och linjeföljning. Den utför AI-uppgifter för att tolka visuell information från omgivningen och fatta snabba beslut om navigering.

- **Arduino Mikrokontroller**

Vi har valt en Arduino-mikrokontroller, ESP8266 alternativt ESP32, som komplement till MaixPy-enheten. Detta är en beprövad och pålitlig mikrokontrollerplattform som är idealisk för motorstyrning och fysisk interaktion med omgivningen.

Arduino-mikrokontrollern är ansvarig för att styra robotens motorer, vilket möjliggör precisionsrörelse och manövreringar. Den kan samordna avståndssensorer och gyroskop för realtidsdatainsamling. Slutligen kommer den hantera kommunikationen med vår centrala server.

- **Motor & Servo**

Vi kommer behöva använda oss av en alternativt två motorer föra att kunna transportera vår robot framåt och bakåt. Sedan kommer vi kunna använda servos för att reglera färdriktningen.

Vår robot kommer förhoppningsvis exklusivt använda motorer och justera färdriktning genom reglerandet av hjul i motsatt riktning. Samma system som bandvagnar och kompaktlastare använder, ibland kallat glidstyrning.

- **Kamera**

För att möjliggöra bildigenkänning och visuell uppfattning kommer vi integrera en högkvalitativ kamera-enhet som är kompatibel med MaixPy SiSpeed M1 Dock.

Kameran fungerar som robotens ögon och fångar detaljerade bilder av omgivningen. Dessa bilder används för att identifiera markeringar, mönster och hinder som roboten måste navigera via.

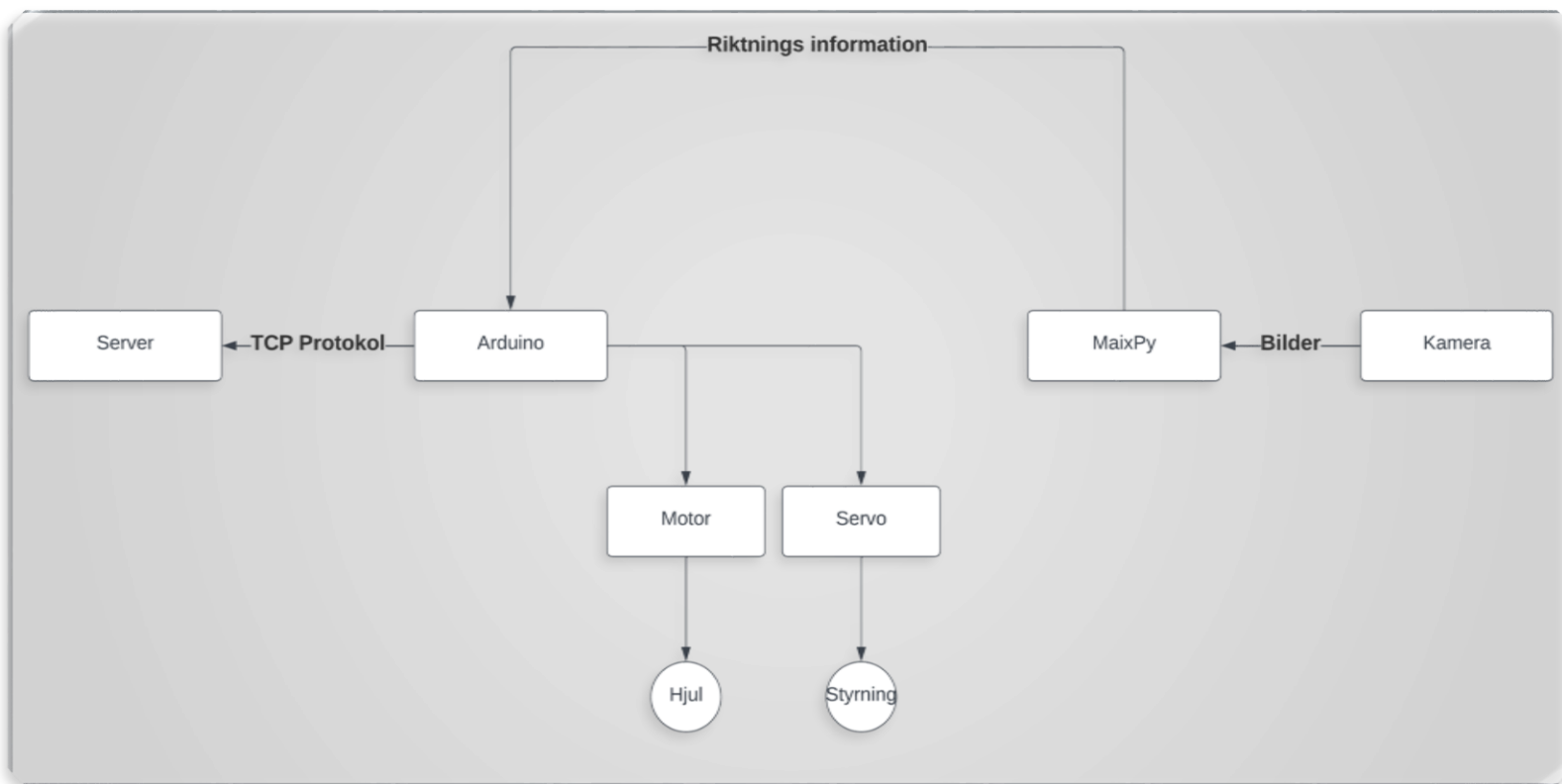
- **Övriga Sensorer**

Vår robot kommer om nödvändigt utrustas med flera sensorer för att förbättra dess förmåga att interagera med omgivningen. Det inkluderar avståndssensorer (ultraljudssensorer), gyroskop, accelerometer och IR-sensorer.

Avståndssensorerna mäter kontinuerligt avståndet till närliggande objekt och möjliggör förhindrande av kollisioner genom att aktivera undvikandemanövrar. Gyroskop mäter robotens rotation och orientering, vilket ger precision i rörelse och navigering.

Vi planerar att ta inspiration ifrån världsberömda Micro-Mouse tävlingen. Micro-Mouse är världsmästerskap inom lösandet av labrynter med helt autonoma robotar. Detta sker på begränsad tid och snabbast lösning vinner.





Ovan är ett mycket enkelt blockdiagram över vår hårdvara.

Beteendebeskrivning

Beteende är ett komplicerat system. Det beskriver handlingar och reaktioner i förhållande till dess omgivning. De centrala system vi upplever att roboten kommer behöva är de följande: Visions Granskande System (VGS), Lokalt Positionerings System (LPS) och Världs Orienterande System (VOS). De olika systemen är alla ansvariga för en viktig del av robotens sysselsättning. De är uppdelade så att de inte bearbetar något mer än en gång. Informationsströmmen är enkelriktad; ingen av de efterkommande systemen kommer skicka information till ett tidigare system.

- **Visions Granskande System**

VGS är ett system som till huvudsak har i arbetsuppgift att ta in kameraflödet och identifiera nya vägbitar. De fyra vägbitarna som kan användas är: raksträcka, sväng, T-korsning och fyrvägs-korsning. I andra hand ser den till att robotens mitt följer markeringslinjen. Det är det två huvudsakliga uppgifterna VGS har.

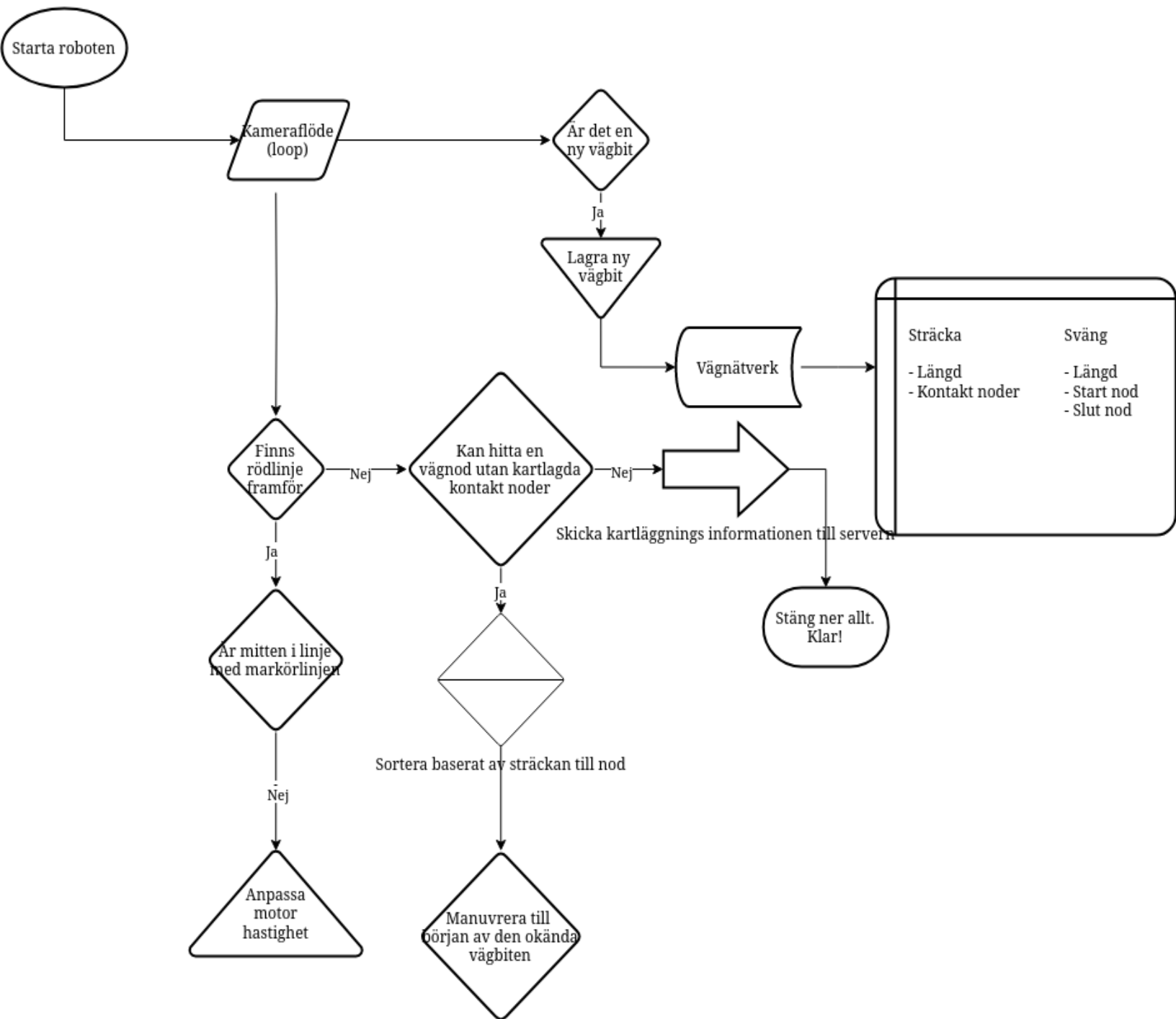
- **Lokalt Positionerings System**

LPS är det system som håller koll på var roboten är i den lokala rymden. Vår lokalrymd är sparad i en speciell datastruktur som tillåter förlängning åt de fyra riktningarna. Till skillnad från vad VGS kan identifiera så har LPS bara koll på två olika vägbitar. LPS har internt koll på sträcka och sväng. För att hålla koll på positionen relativt till vart den startade använder den

distansmätningar (Odometry) med den tidigare nämnda accelerometern. Då distansmätning på det viset kan vara felbenäget innebär det att vi behöver kunna arbeta med approximativa värden där felet är en okänd variabel som är varierande utöver systemet.

- **Världs Orienterande System**

Kompletterande till LPS har vi VOS. VOS är det system som tar hand om orienteringen av roboten. Den har full kontroll över motorerna vilket innebär att det är den som styr. För att roboten ska kartlägga hela vägnätverket krävs det ett samarbete mellan alla våra system. VOS tar hand om att roboten följer markörlinjen, tar sig till icke kartlagda kanter och att hantera regleringssystemet. Inget av systemen är mer viktigt än ett annat men VOS är det system som gör allt det synliga arbetet i vår värld.



Deluppgifter*

1. Hårdvarusystem: konfigurera och integrera robotens hårdvara

Planering och identifiering:

- Välj och förvärva de nödvändiga hårdvarukomponenterna, inklusive MaixPy M1S, motorer, servo, kamera och övriga sensorer.

- Översiktlig design av robotens fysiska konfiguration, inklusive placering av komponenter.

Implementering och testning:

- Bygg den fysiska roboten.
- Anpassa MaixPy M1S och Arduino mikrokontroller.
- Testa robotens fysiska rörelse och säkerställ att allt fungerar.

2. Bildigenkänning: utveckla algoritmer och träning

Planering och identifiering:

- Utvärdera olika bildigenkänningsalgoritmer, inklusive OpenCV, TensorFlow, YOLO osv.
- Samla och förbered ett omfattande dataset av bilder.
- Träna valda algoritmen med det förberedda materialet och säkerställ dess förmåga att identifiera vår väg.

Implementering och testning:

- Integrera den tränade systemet med robotens hårdvara och mjukvara.
- Skapa ett gränssnitt för att överföra information från bildigenkänningsystemet till linjeföljningsalgoritmen.
- Testa systemet i olika miljöer och justera efter behov och prestanda.

3. Linjeföljning: utveckla algoritmer för precision

Planering och identifiering:

- Designa algoritmer för att följa linjer med precision, inklusive hur roboten reagerar på avvikelser från banan.
- Planera för hur roboten ska justera sin rörelse för att korrigera eventuella avvikelser och förhindra avvikelser i framtiden.

Implementering och testning:

- Integrera linjeföljningsalgoritmen med robotens motorer och styrsystemet.
- Skapa en mekanism för att ta emot information från bildsystemet och använd den för att styra robotens rörelse.
- Testa linjealgoritmen på olika typer av banor och justera den för att uppnå precision.

4. Digitalt vägnätverk: skapa och hantera kartinformation

Planering och identifiering:

- Utforma en struktur för det digitala vägnätverket som kan användas för att representera området och positionen.
- Planera hur informationen om robotens plats och eventuella hinder kommer att lagras och uppdateras.

Implementering och testning:

- Implementera det digitala vägnätverket i robotens mjukvara och hårdvara.
- Skapa gränssnitt för att generera och uppdatera kartinformationen baserat på robotens nuvarande variabler.

5. Kommunikation med server: skapa och underhåll fjärranslutning

Planering och identifiering:

- Definiera de nödvändiga protokollen med tekniker för kommunikation mellan roboten och server.
- Planera hur data ska överföras, hur ska roboten kommunicera sina framsteg och eventuella problem.

Implementering och testning:

- Implementera kommunikationsprotokollen och teknikerna i robotens mjukvara på servern.
- Skapa en säker och tillförlitlig anslutning mellan robot och server, med hänsyn till eventuella nätverksproblem och eventuella förluster av data.
- Testa kommunikationssystemet genom att låta roboten skicka data och ta emot kommandon från servern.

6. Förflyttning och styrning: skapa robotens rörelsemönster

Planering och identifiering:

- Designa robotens rörelsemönster och beteende, inklusive hur den ska accelerera, bromsa och svänga.
- Planera för att roboten ska reagera på externa händelser och kommandon, som stoppsignaler eller nödsituationer.

Implementering och testning:

- Implementera rörelsemönstren och styrningsarbetet i robotens mjukvara och koppla dem till motorer och servon.
- Testa robotens rörelsemönster och säkerställ att den kan följa en bana och svara på styrkommandon korrekt.

7. Övriga sensorer: integrera och använd andra sensorer

Planering och identifiering:

- Utvärdera behovet av ytterligare sensorer, som avståndssensorer, gyroskop, accelerometer, IR-sensorer för att förbättra robotens förmåga att interagera med omgivningen.
- Planera hur dessa sensorer kommer att användas och integreras i robotens beteende.

Implementering och testning:

- Integrera ytterligare sensorer i robotens hårdvara och mjukvara.
- Skapa beteende och algoritmer som använder sensorinformationen för att undvika hinder, mäta avstånd och förbättra navigeringen.
- Testa sensorernas funktion.

Tidsplan

Förberedelsefas (2 veckor)

- Vecka 1–2 (13–26 september): Projektdefinition och planering
 - Definiera projektets omfattning och mål
 - Identifiera och förvärva de nödvändiga hårdvarukomponenterna
 - Organisera projektgruppen och fördela roller och ansvar
 - Förberedd inför opponering
 - Påbörja projektet

Utvecklingsfas (13 veckor)

- Vecka 3–4 (27 september – 10 oktober): Hårdvaruintegration och förberedelse
 - Bygg robotens fysiska plattform
 - Installera och konfigurera MaixPy SiSpeed M1 Dock och Arduino-mikrokontroller
 - Förbered tidigare utvecklat TCP protokoll
 - Förbered inför Milestone 1
- Vecka 5–8 (11 oktober – 7 november): Bildigenkänning, förflyttning och styrning (Milestone 1: 8 november)
 - Implementera rörelsesystemet och motorstyrning
 - Implementera och testa bildigenkänningssystemet
 - Förbered det initiala bildigenkänningsdatasetet
 - Få roboten att identifiera objekts förflyttning
 - Förflytta roboten efter identifierat objekt
 - Enkel kommunikation med server för objektföljning
 - Om tid finns påbörja linjeföljningsalgoritm
- Vecka 9–12 (8 november – 5 december): Digitalt vägnätverk, kommunikation och identifiera linjer
 - Implementera linjeföljningsalgoritmer och integrera dem med hårdvaran
 - Implementera digitalt vägnätverk och kartuppdateringssystem
 - Fortsätt utveckla kommunikationsprotokoll för att skicka data till servern
 - Förbered inför Milestone 2
- Vecka 13–15 (6 december – 26 december): Legovägs följning (Milestone 2: 13 december)
 - Följ rak legoväg
 - Följ svängande legoväg
 - Följ rakt vid korsande legoväg
 - Optimera och förbättra robotens förmåga att förflytta sig och svänga

Implementeringsfas (4 veckor)

- Vecka 16–17 (27 december – 9 januari): Navigera legoväg
 - Öka precisionen vid linjeföljning
 - Utveckla vägnavigation (inte bara följa)
- Vecka 18–19 (10 januari – 23 januari): Digitalt vägnätverk och Kartläggning

- Implementera robotens förmåga att skapa ett digitalt vägnätverk
- Skapa och uppdatera kartor över områdena som roboten utforskat

Avslutningsfas (4 veckor)

- Vecka 20–21 (24 januari – 6 februari): Kommunikation med server och datahantering
 - Slutför kommunikationsprotokollen med servern
 - Om nödvändigt implementera datalagrings- och visualiserings-funktioner
 - Påbörja slutgiltiga rapporten
 - Förbered inför kursens slut
- Vecka 22–23 (7 februari – 20 februari): Dokumentation och redovisning (Examination: 14 februari, Projektrapport: 21 februari)
 - Genomför de sista tester och optimeringar
 - Förbered inför examination
 - Slutför projektplanen och rapporten

mdu dval38

Redovisningar

- Milestone 1
- Milestone 2
- Examination
- Projektrapport

Förberedelsefas

Projektdefinition

- Definiera projektets omfattning och...
- Identifiera och förvärva de nödvänd...
- Organisera projektgruppen och för...
- Förbered inför opponering
- Påbörja projektet

Utvecklingsfas

Hårdvaruintegration och förbere...

- Bygg robotens fysiska plattform
- Installera och konfigurera MaixPy S...
- Förbered tidigare utvecklat TCP pro...

Bildigenkänning, förflyttning och...

- Implementera rörelsesystemet och...
- Implementera och testa bildigenkän...
- Förbered det initiala bildigenkänn...
- Få robot att identifiera objekts förl...
- Förflytta robot efter identifierat obj...
- Enkel kommunikation med server för...

Digitalt vägnätverk, kommunikation...

- Implementera linjeföljningsalgorit...
- Implementera digitalt vägnätverk ...
- Fortsätt utveckla kommunikationsp...

Legovägs följning

- Följ rak legoväg
- Följ svängande legoväg
- Följ rakt vid korsande legoväg
- Optimera och förbättra robotens fö...

Implementeringsfas

Navigera legoväg

- Öka precisionen vid linjeföljning
- Utveckla vägnavigation (inte bara f...

Implementera robotens förmåga...

- Implementera robotens förmåga ...
- Skapa och uppdatera kartor över...

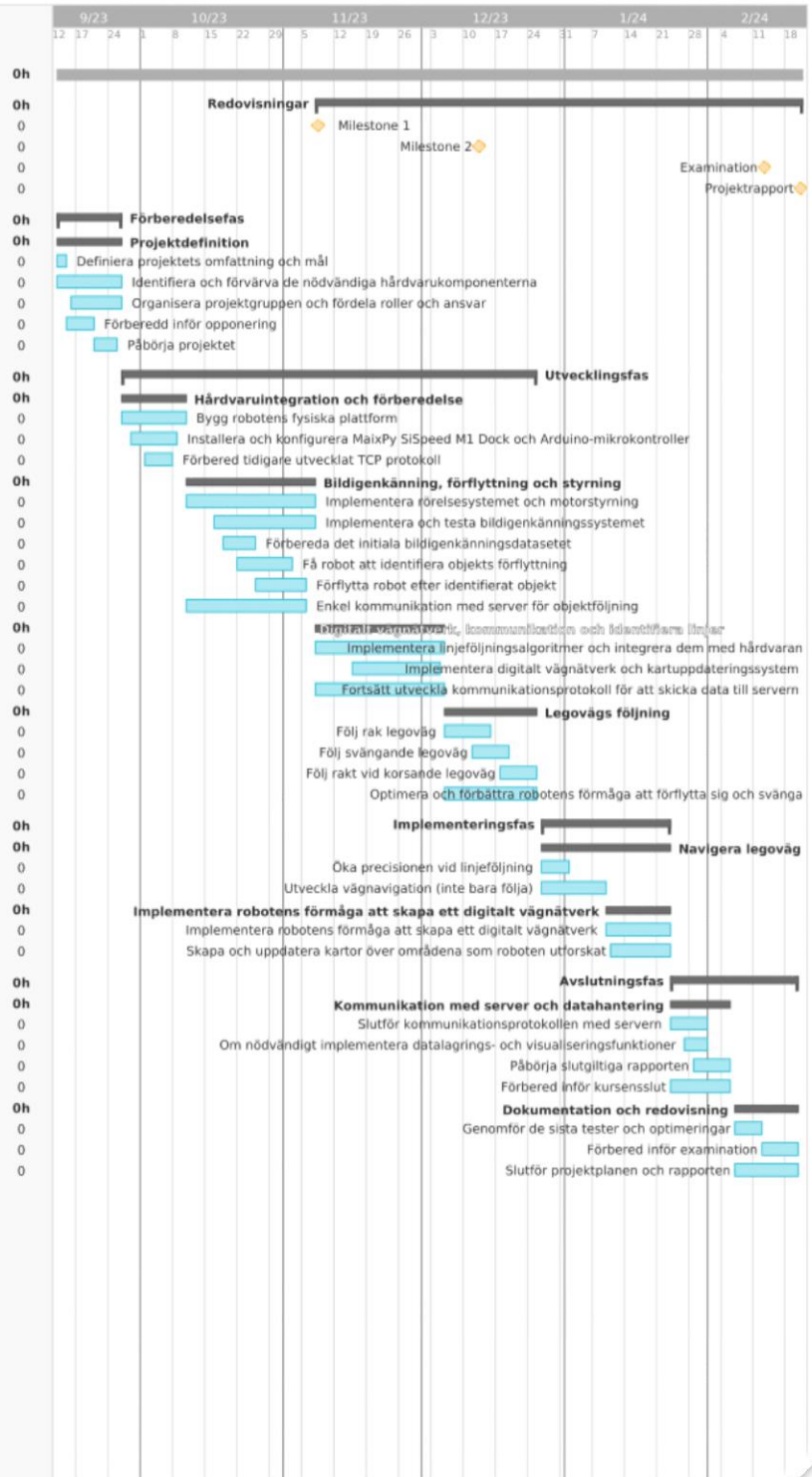
Avslutningsfas

Kommunikation med server och ...

- Slutför kommunikationsprotokollen...
- Om nödvändigt implementera data...
- Påbörja slutgiltiga rapporten
- Förbered inför kursens slut

Dokumentation och redovisning

- Genomför de sista tester och opti...
- Förbered inför examination
- Slutför projektplanen och rapporten



*Kommande siffror har inte någon betydelse när det kommer till vilken ordning de ska utföras. Se tidsplan för det.