

Teknisk Rapport IN1060 – Bruksorientert design

Veles presenterer en oppgave av:

Anna Hansgård,
Isabelle Røed Lampl,
Linn Huserbråten,
Mari Notland,
Shannen Leigh,
Viljar Drevland Hardersen.



31.05.23

Innholdsfortegnelse

1. Introduksjon	3
1.1 Målgruppe	3
1.2 Tema og problemstilling	3
1.3 Konsept	3
1.4 Motivasjon	3
1.5 Vår løsning	4
1.6 Krav	4
2. Utforsking, eksperimentering og konstruksjon	5
2.1 Teknologien	5
2.1.1 HC-SR501 Passiv infrarød sensor	5
2.1.2 HC-SR04 Ultrasonisk sensor	6
2.2 Konstruksjon	7
3. Sluttprodukt	8
3.1 Artefakten	8
3.2 Komponenter brukt	9
3.3 Kretstegning	11
3.4 Kode	11
3.4.1 Variabler og konstanter	11
3.4.2 Metoder	12
3.4.3 Kodens oppbygging og løp	13
4. Konklusjon og tanker	14
4.1 Design	14
4.2 Begrensninger	14
4.3 Fremtiden	14
5. Film	15
6. Referanseliste	15

1. Introduksjon

1.1 Målgruppe

Målgruppen vi bestemte å ta for oss ble til slutt *voksne og erfarne sjåførere* som har hatt førerkortet i over fem år, kjører jevnlig i forbindelse med jobb og ferie, og anser seg selv som erfarne og trygge førere.

1.2 Tema og problemstilling

Temaet vi tar utgangspunkt i er dyrepåkørsler, mer spesifikt hjortevilt. **Problemstillingen** vi prøver å løse er *hvordan redusere antall og omfang av dyrepåkørsler på særlig utsatte strekninger*.

1.3 Konsept

Konseptene vi jobbet ut fra er *varsel og oppmerksomhet* – Vi ønsker å gi brukeren tidlig varsel og gjøre de oppmerkssomme på hjortevilt i nærheten. **Formkonseptet** vi designet etter er *mobil* – Noe bærbart og kjent og kjært for de fleste, vi ønsker ikke at vår løsning skal stikke seg ut eller være distraherende med mindre noe skal varsles.

1.4 Motivasjon

Dyrepåkørsler, spesielt påkjørsel av storvilt er et stadig voksende problem (Hjorteviltregisteret.no, u.å.) som fører til store skader på folk, dyr og materiell. Etablerte løsninger er meget kostbare og bringer andre problemer, mens alternative tilnærminger for å forhindre påkjørsler har ikke hatt beviselig påvirkning (Wildenschild, 2022, s. 34.). Vår brukerbase består av erfarne førere som ofte ferdes på utsatte strekninger og kjenner folk som har vært involvert i ulykker. Med deres hjelp ønsker vi å skape noe litt mer elegant som passer inn i deres hverdag og deres type kjøring.

Måten vi tenker vår løsning skal bidra til å løse problemstillingen er å sanse det vi ikke kan sanse, se det vi ikke kan se før det er for sent. Hjortevilt er godt tilpasset et liv i skogen og kamufleres godt mot bakgrunnen – Vi ønsker å bruke sensorer for å trekke de frem, ikke for alle å skue, men for å kunne gi presis varsling i god tid slik at kjøreatferd kan justeres og påkjørsler unngås.

1.5 Vår løsning

Etter masse idémyldring, skissering, workshopping og prototyping kom vi omsider frem til en løsning både vi og våre brukere var fornøyde med – En innretning i bilen med et meget enkelt og konsist varslingsystem som blinker og piper når sensoren som er plassert foran og på siden av bilen merker at det er noe som beveger seg i nærheten.

1.6 Krav

Da idéen til hvordan den faktiske artefakten skulle se ut begynte å ta form måtte vi naturligvis utarbeide krav til hva slags interaksjon den skal tilby, hva slags informasjon den skal servere og hvordan. Sistnevnte, altså *ikke-funksjonelle krav* blir noe tynt i denne omgangen da vi er noe begrenset av Arduino sine muligheter i tillegg til at vårt fokus i hovedsak var på muligheter.

Tabell 1

Oversikt over funksjonelle krav

Krav	Utdypning
Blinkende rødt lys for varsel	Brukerne ønsket kun én pære som skulle varsle, og den vil blinke uansett om sensoren plukker opp noe i nærheten, om trykker knappen for å varsle, eller at den plukker opp et varselsignal fra andre artefakter.
Lydvarsling	Slik at artefakten piper i tillegg til blinkende lys
Mulighet for å justere volum på lydvarsling **DELVIS IMPLEMENTERT**	Lett tilgjengelig volumbryter for å skru av selve lyden. Piezoen som følger med Arduinosettet tillater ikke justering av volum, så bryteren, altså potensiometeret, har kun av eller på.
Bryter for å skru artefakt av og på	Lett kjennbar knapp med tydelig klikk for at artefakt er av og på.
Lys for å vise at artefakt er av og på	Synlig, men ikke inntrengende lys som gir tegn på om artefakten er av eller på.
Mulighet til å feste artefakten i bilen	Biler er forskjellige, brukerne må ha flere festemuligheter slik at den er best tilgjengelig og synlig i deres bil.
Artefakten skal være lett å operere med én hånd uten å se på	Vi ønsker ikke at artefakten skal ta for mye koordinering eller oppmerksomhet fra kjøringen

Knapp for å varsle andre	En tydelig og lett tilgjengelig knapp som setter i gang varslingen over nettverket.
Artefakt koblet til nettverk av lignende artefakter **IKKE IMPLEMENTERT**	Ønsketenking at artefaktene er koblet til et nettverk gitt en strekning slik at de som er lengre unna kan varsles om en bruker oppdager noe og ønsker å varsle andre brukere.

Tabell 2

Oversikt over ikke-funksjonelle krav.

Krav	Utdypning
Varsleknapp skal holdes inne to sekunder	Holdes inne over litt tid slik at man ikke kommer bort i den ved et uhell
Skal varsles i 30 sekunder	Høvelig varsling, men ikke for lenge
Skal varsle dyr innen 200 meter **IKKE IMPLEMENTERT**	Etter diskusjon med ekspert ble det foreslått at 200 meter var grei avstand å varsle innen, men grunnet tekniske begrensinger var det ikke gunstig å prototype for det i denne omgang.

2. Utforsking, eksperimentering og konstruksjon

2.1 Teknologien

Utgangspunktet i oppgaven vår er *sensorer* – vi ønsker å *sanse* noe, og Arduino sitt rammeverk tilbyr oss to relevante sorter – Ultrasonisk (Radar) og infrarød. Disse kommer i mange varianter og prisklasser, men av økonomiske og tidsrelaterte hensyn gikk vi for det som var lettest tilgjengelig og eksperimenterte med en HC-SR04 ultrasonisk sensor og en HC-SR501 PIR (Passive InfraRed) sensor.

2.1.1 HC-SR501 Passiv infrarød sensor

I utgangspunktet virker infrarødt som et åpenbart valg for å detektere store varmesignaturer på avstand da dette brukes blant annet til jakt av storvilt. Ulempen er at disse er svært kostbare, så for våre formål må de utgå selv om de kunne vært meget relevante for et ferdig produkt. Noe litt mer tilgjengelig er en *passiv infrarød sensor* som detekterer en varmesignatur innenfor sitt synsfelt og varsler. Slike brukes blant annet til automatiske lysbrytere som aktiveres i det en bruker kommer inn i rommet. Tanken var at vi kunne plassert noe slikt utenpå bilen vendt mot veikanten og at den varsler når det kommer en varmesignatur, men vi endte med å *ikke* gå for denne da den var *meget* sensitiv, og

gjorde utslag omtrent uansett hvor en befant seg. Den var litt for grov, og tross begrensningene våre ønsket vi noe litt mer smidig og med høyere oppløsning om vi skulle få behov for det.

Figur 1

PIR-sensor



Notat. Med og uten kuppel. Den detekterer varmesignaturer innen 15 meter og en kjegle på 90 grader.

2.1.2 HC-SR04 Ultrasonisk sensor

Figur 2

Ultrasonisk radar



Notat. Radar hvor ene siden sender ut lydssignal, andre siden mottar og signaliserer til Arduino. Bak ser du graf som viser avstand.

Radaren funker som de har gjort siden krigen – Lydgiveren (TRIG) sender ut et lydssignal som reflekteres av et objekt, og mikrofonen (ECHO) tar imot og sender et signal vi behandler med Arduinoen. Vi endte med å benytte oss av denne av flere årsaker. Den tilbød oss mer fleksibilitet samtidig som den i praksis kunne tilby samme funksjonalitet som PIR-sensoren. Med denne kan vi måle avstand fremfor å bare varsle innen PIR-sensoren sin fastsatte, og tross sine 15 snevre grader med radius kan den monteres på en servo slik at den vrir seg frem og tilbake. Siden vi vet

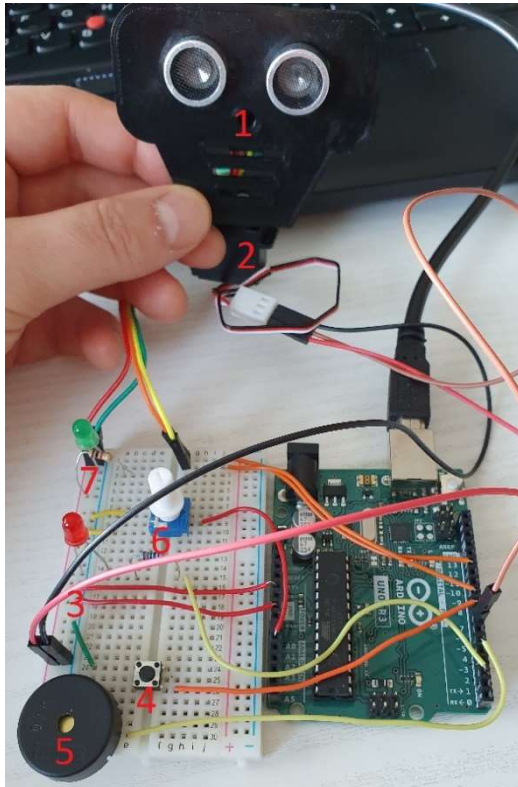
vinkelen på servoen vil vi med avstandsmåling vite nøyaktig hvor et objekt befinner seg i forhold til sensoren. Etter workshopping med brukere endte vi opp med å ikke bruke all denne dataen da vi kom frem til at for mye informasjon kun vil bidra til forvirring, men vi holdt oss til radaren da den var meget effektiv for prototypingsformål og betraktelig mindre kranglete enn PIR-sensoren.

2.2 Konstruksjon

Med funksjonelle krav bestemt og sensor valg satte vi i gang med å bygge prototypen opp modul for modul, delt noenlunde som følger:

Figur 3:

Elektronikken prototypet



1. Radar – Bedømme avstand
2. Servo – Styrer sensor, bedømme vinkel
3. Lysvarsel – Lyssignal for å varsle bruker
4. Varsleknapp – Lar bruker varsle andre
5. Lydvarsel – Lydsignal for å varsle bruker
6. Lydkontroll – Bryter som lar bruker justere lyd
7. Lys for om enhet er av eller på

I tillegg skal det være en bryter for å skru enheten av og på, men denne kobles i serie med batteriet.

Selv om vi har lært disse å kjenne i løpet av kurset var det en utfordring å få alt til å «snakke sammen» og kjøre i tandem grunnet utfordringer i å få Arduino til å gjøre flere ting samtidig. Et problem vi ikke har klart å fikse ennå er at radaren bruker ekstra mye tid når den skal registrere ting langt unna, og sakter ned hele systemet som følge, inkludert roteringen.

3. Sluttprodukt

3.1 Artefakten

Etter mye kjemping mot lite plass, en Arduinoenhet som kortsluttet, kabler som var for stive slik at loddingen knakk, og slarkete koblinger fikk vi trykket alt inn i en boks. Denne passer i en generisk mobilholder som kan plasseres i bil hvor bruker selv ønsker. Den tilbyr grensesnitt som beskrevet oven. Sensoren ville vi ideelt plassert foran til høyre på bilen slik at radaren kan sjekke foran bilen og 90 grader til høyre for å dekke det kritiske feltet. Dette styres automatisk og brukeren vil stort sett bare forholde seg til selve varslingsenheten.

Figur 4

Varslingsenhet på display (over) og montert i bil (nedenfor)

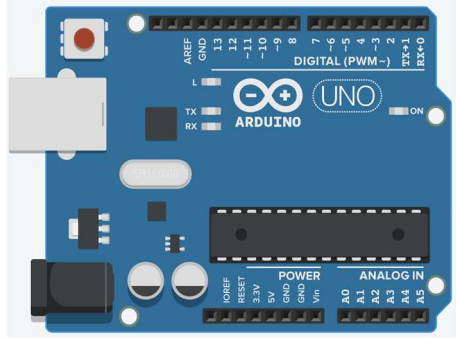
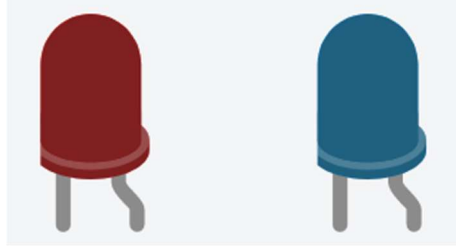
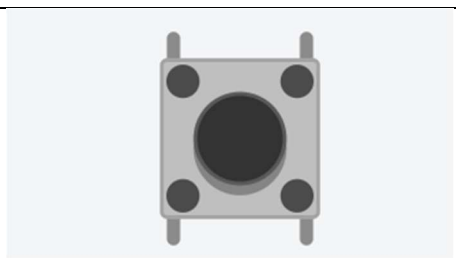
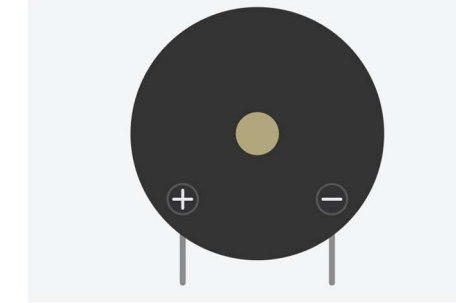
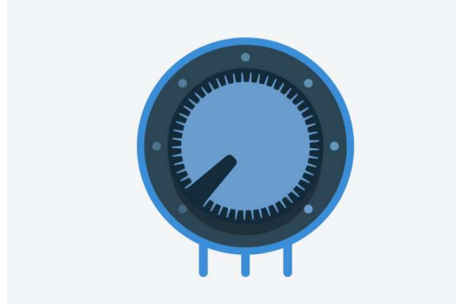


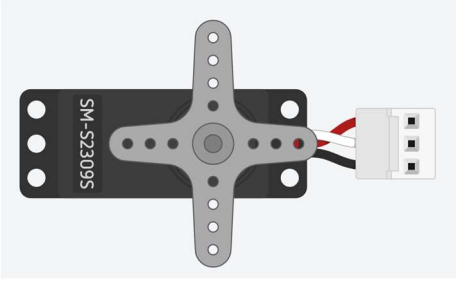
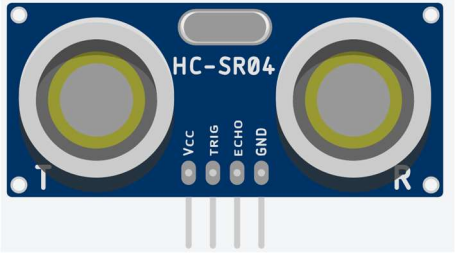
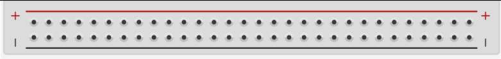
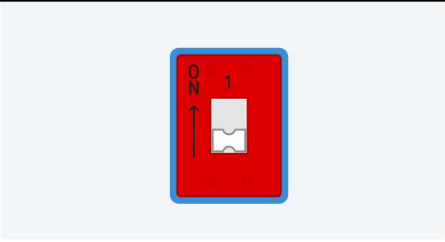
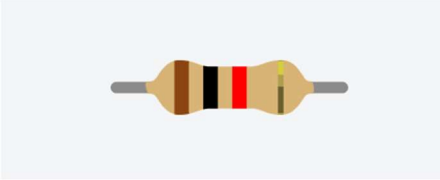
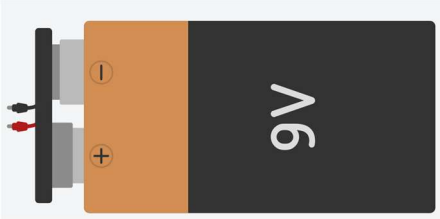
Notat. Nederst ser du også radaren på servo, men den ville i et ferdig produkt vært montert utenpå.

3.2 Komponenter brukt

Tabell 3

Oversikt over komponenter

Komponent	Illustrasjon	Funksjon	Antall
Arduino Uno		Programmerbar mikrokontroller som blir gitt et sett instruksjoner å utføre gitt input fra bruker og sensorer.	1
LED (rød og blå)		Lyser opp når de får strøm gjennom seg – Rød brukes til å varsle bruker om fare, blå til å varsle at enhet er på.	1 blå 2 røde i parallell for å få nok lysstyrke
Knapp		Slutter kretsen rundt hele når trykket inn – Brukes til å aktivere varslefunksjonen i systemet.	1
Piezo		Lager lyd når den får strøm gjennom seg, tone bestemt av frekvens. Gir lyd når systemet varsler og lydbryter.	1
Potentiometer		Gir input mellom 1 og 1023 basert på bryterens posisjon. Gir bruker mulighet til å bestemme om det skal varsles med lyd i tillegg til lys.	1

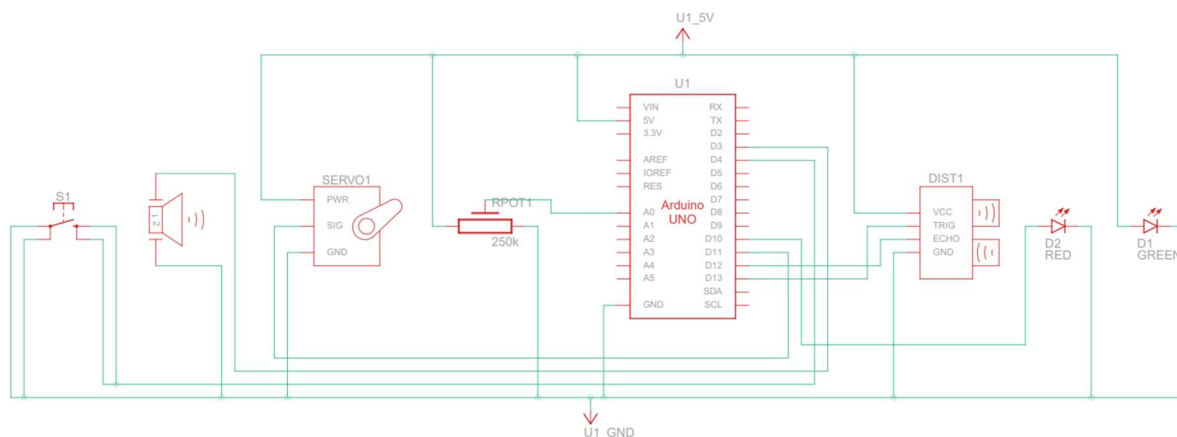
Servo		Roterer til gitt vinkel basert på input fra Arduino. Brukes til å rotere radaren for å få bredere synsfelt.	1
Ultrasonisk avstandsmåler (Radar)		Har lydelement som sender ut puls og mikrofon som plukker den opp. Via Arduino kan vi måle tiden dette tar og bedømme avstand.	1
Strømforsyning (Demontert breadboard)		Kobles til 5V og jording i Arduinoen for å forsyne de andre komponentene med strøm.	1
Bryter		Strømbryter koblet i serie med strømforsyningen til Arduinoen – Skruer alt av og på.	1
Resistor		Øker resistansen i seriekoblingen – Demper LED-en.	1
Batteri		Gir strøm til Arduinoen, gjør artefakten trådløs.	1
Kabler	-	Koble alt sammen	~30

Notat. Illustrasjoner hentet fra Tinkercad under CC-lisens.

3.3 Kretstegning

Figur 5

Kretstegning laget i Tinkercad



3.4 Kode

Kodefilen er lagt ved innleveringen som «Radarvarsler.ino» og er naturligvis linket til på nettsiden vår. Bruken av globale variable beklages på forhånd.

3.4.1 Variabler og konstanter

Tabell 4

Oversikt over programmets konstanter og variabler, i tillegg til deres formål.

Navn	Formål
duration	Arduino har en innebygget metode vi kommer tilbake til, men vi bruker den til å bestemme denne verdien som sier hvor lenge lydimpulsen var i luften. Gis pr 10 μ s.
dist	Ovennevnte brukes til å kalkulere denne som er distansen i cm til objektet lydimpulsen ble reflektert av.
potV	Verdien Arduinoen leser av potensiometeret, brukes til å bestemme om varslingen skal gi lyd.
pos	Bestemt av verdien til potmeteret – Har her bare verdi 0 og 1 for av og på.

<code>varsleDist</code>	Konstant hvor vi angir i cm hvor nærme noe kan være før varslingen skal slå inn.
<code>varsleTid</code>	Konstant hvor vi angir hvor lenge det skal varsles i sekunder.
<code>servoA</code>	Variabel som forteller hvilken vinkel i grader servoen skal stå i.
<code>direction</code>	Boolsk verdi som bestemmer om servoen skal øke eller minske vinkelen, altså vri radaren mot høyre eller venstre.
<code>buttCount</code>	Teller som holder styr på hvor lenge varsleknappen er holdt inne slik at man ikke bare dulter borti og varsler alt.
<code>servis</code>	Servoobjektet vi skriver vinkelverdi til

3.4.2 Metoder

radarPulse()

Til stor hjelp var det at radarmodulen er godt dokumentert, og i ressursene som ble brukt til å utforske dens funksjon ble det sterkt anmodet å bygge radarpulsene som det ble gjort i koden, og den opererer i tre steg: Først sørger den for at det ikke sendes lyd i 5 μ s, så sender den et høyfrekvent lydsignal som varer i 10 μ s, og til sist skrur den lydmodulen av. Dette sørger for et tydelig og rent signal. Den jobber på trigger pin, altså lydenheten på radarmodulen og sender et høyfrekvent lydsignal.

pulseIn(echoPin, HIGH)

Dette er en innebygget funksjon i Arduino sitt utviklingsmiljø som gjør radaren meget brukervennlig. Forrige funksjon virket på trigger pin, denne virker på echo pin, altså mikrofonen som tar imot lydsignalet. Som formulert i koden sjekker den hvor lang tid i 10 μ s det går mellom den veksler mellom HIGH og LOW – Altså hvor lang tid lyden den sendte bruker frem og tilbake.

distCalc()

Denne metoden tar for seg ovennevnte og kalkulerer avstanden til objektet pulsen reflekteres av. Vi lagrer tiden i variabelen `duration` og deler den på 100 for å få den over i ms heller. Deretter halverer vi tiden fordi vi er kun interesserte i hvor lang tid lyden brukte én vei. Som vi lærte i fysikken er hastigheten til noe avstanden den reiste over en gitt tid, og takket være `pulseIn` vet vi nå to av tre. Vi

anslår lydens hastighet til å være 343 m/s (ved 20 grader C og tørr luft), men fordi vi ønsker distanse i cm og har tiden i ms må vi dele hastigheten på 100. Funksjonen returnerer da distansen målt til objektet i cm.

volumKontroll()

Denne funksjonen leser av verdien til potensiometeret og returnerer 1 eller 0 som vil bli brukt i varslingsmetoden til å bestemme om den skal ha med lyd eller ikke. I en idéell verden hadde den returnert en verdi som skulle bestemt volumet til varslingen, men her har den bare av eller på.

varsle(int dist)

Denne metoden tar imot en avstand og sjekker om den er innenfor terskelen vi har satt for når det varsles. I så fall setter den i gang en løkke som varsler i tiden vi anga tidligere i programmet. Lyset blinker et halvt sekund på og halvt sekund av, og leser av potensiometeret via ovennevnte metode for å sjekke om den skal gi lyd i tillegg.

3.4.3 Kodens oppbygging og løp

Med vår kunnskap om variabler og metoder kan vi nå ta en gjennomgang av hva koden faktisk gjør og hvordan det bidrar til å løse problemene og kravene vi har satt.

Metoden `setup()` kobler Arduinoen til komponentene slik at den vet hva som er hvor, i tillegg til at den ber piezoen gi et pip for å demonstrere at enheten er startet og at volum er på.

Deretter starter `loop()` som repeteres så lenge enheten er på. Den starter med å sjekke hvorvidt varsleknappen holdes nede, og fordi vi bruker `INPUT_PULLUP` vil den i så tilfelle lese LOW. Da vil den øke telleren `buttCount` for hver gjennomgang av løkka med én, for så å sjekke om den har nådd grensen vi har satt med hvor lenge knappen skal holdes inne. Å time dette sånn at den skal holdes nøyaktig to sekunder er ganske vanskelig fordi det er variabelt hvor lang tid radaren bruker på å hente signalet, men en teller på 60 ga oss cirka to sekunder på det jevne. Vi kunne i teorien brukt `millis()`, men dette hadde løst noe som knapt kan kalles et problem, og medført andre komplikasjoner. Når telleren har nådd terskelen vil den kalle på `varsle()` og sende inn 0 som «målt avstand» da dette alltid vil utløse en varsling. Hvis knappen slippes før det har gått to sekunder resettes telleren til 0 slik at skulle man være veldig klønete og dulte borti knappen ofte vil den ikke utløse alarmen. Når dette er gjort kalles `radarPuls()`, og vi henter variabelen `dist` som tar imot verdien fra `distCalc()`, altså avstanden til objektet radaren ser. Denne avstanden blir sendt til metoden `varsle()` som sjekker om avstanden er innenfor terskelen vi har satt, og hvis den er det slår varslefunksjonen ut. Når alt dette er gjort sjekker den hvilken retning radaren skal bevege seg, og øker eller minsker vinkelen der etter. Hvis den har nådd vår satte terskel på 90 grader endrer den retning, og endrer retning tilbake når den når 0.

4. Konklusjon og tanker

4.1 Design

Vi prøvde etter beste evne å etterkomme brukernes ønsker når det kom til utforming og valg av farger, og vi er meget fornøyde med sluttresultatet. I totalt endte vi med å ha 3D-printet tre bokser, to av hvis ble destruert for eksperimentering for å få til knapper og brytere, men også for å få plass til alt vi trengte inni. Selv om PIR-sensorene ikke ble brukt tok vi beskyttelseskuppelen og 3D-printet et feste til de to røde LED-lysene som blir brukt til å varsle.

4.2 Begrensninger

Det vi i hovedsak har prototypet i denne omgang er hvordan det oppleves for brukeren å ha en innretning som dette i bilen, i tillegg til oppsett, utforming, farger og materialer. I brukertesting var selve sensoren ikke veldig sentral på grunn av komponentenes begrensninger og manglende tilgang på stødig forsyning av storvilt i veikanten.

Vi skulle gjerne ønske at selve sensoren var trådløs, noe som ville krevd bruk av Bluetooth. Vi anså ikke dette som verdig bruk av vår tid for denne prototypen, men er definitivt noe vi ville vurdert i en mer avansert variant.

Som tidligere nevnt sliter radaren med å prosessere signal hvis ting er lengre unna enn 3-4 meter, som fører til at servoen ikke går videre før den har sjekket om det er noe innenfor varslingsavstand. Dette er ikke idéelt da ting skjer meget fort når det kommer til dyrepåkjørslar.

En annen svakhet er at vår idé ikke ville funket med svingete veier, tett skog, skrenter og lignende. Planen der er at varsleknappen ville funket som en slags failsafe, men på vei til hytta langt opp i berget er det ikke sikkert det er mange som kjører. Dette måtte føreren naturligvis tatt i betraktning.

Løsningen krever også at sensoren pålitelig kan detektere storvilt, ellers er vi like langt.

4.3 Fremtiden

De fleste moderne biler er allerede utstyrt med en form for sensor i front og har en ganske avansert datamaskin brukeren interagerer med. Vi ser for oss at idéen av løsningen vår kunne vært koblet til denne sensoren og vært integrert i systemet selv.

Ellers blir sensorteknologi mer avansert og tilgjengelig, men spesielt i nåværende tidsalder hvor kunstig intelligens har blitt tilgjengelig for allmennheten kunne formgjenkjenning vært aktuelt. Vi er klare over at løsninger som har implementert høyoppløselig radar med formgjenkjenning eksisterer, men de er fortsatt på infrastrukturnivå og meget kostbare.

5. Film

Link til film: https://drive.google.com/file/d/1347adlMjS6HMuo1s3wxZJEnTZ_KHeTNw

Filmen åpner med å vise en rulletekst med eksisterende tiltak, permanente og eksperimentelle, for så å gå gjennom to scenarioer oppbrutt av en liten teknisk forklaring. Først ser vi at føreren, distraheret av høy musikk og dårlig tid, ikke får noen forvarsel på at det er et dyr i nærheten som vil ut på veien. Dette ender med en påkjørsel av dyret, det siste vi ønsket skulle skje. Etter dette kuttet det til en kjapp gjennomgang og demonstrasjon av artefaktens teknologi og funksjoner, før vi til sist viser den i brukskontekst. Elgen ble av en eller annen grunn ikke fanget opp av sensoren før brukeren så den, så bruker sakkett ned og aktiverte varslingsystemet ved å holde knappen ned. En annen bruker i nærheten med lik enhet fanger varselet, og justerer kjøringen deretter. Dette forhindrer dermed ulykken som skjedde i første scenario, og alle sammen kom trygt frem til dit de skulle.

6. Referanseliste

Hjorteviltregisteret. (u.å.). Statistikk om fallvilt, lest april 2023 fra

<https://hjorteviltregisteret.no/FallviltInnsyn/Statistikk/>

Wildenschild, H. (2022). Statens vegvesen sine forsøk for å redusere antallet viltpåkjørsler, s34.

<https://vegvesen.brage.unit.no/vegvesen-xmlui/bitstream/handle/11250/2985804/SVV%20rapport%20nr.%20803%20Viltp%C3%A5kj%C3%B8rsler.pdf>