

Teknisk Rapport

Effektivisering av natteravnene i Oslos ressurser

Kristine Lokke

Lotte Stubdal

Renate Farberg

Ane Marthe Ness

Anna Helgestad



uRO

VI SKAPER RO I UROEN

Prosjektoppgave i IN1060 Institutt for Informatikk

UNIVERSITETET I OSLO

12.06.2019

Innholdsfortegnelse

INTRODUKSJON	2
1. OM OSS	2
2. MÅLET MED LOGGFØREREN	2
3. PRESENTASJON AV PROSJEKTVIDEO	2
4.1 BYGGING	5
4.2 UTFORDRINGER MED KOMPONENTENE	6
5.1 FORKLARING AV KODE	7
5.2 GPS SHIELD	9
6. HVORDAN HAR KRAVENE BLITT NÅDD?	9
7. VEIEN VIDERE	10
8. KILDELISTE	12

Introduksjon

I denne rapporten presenterer vi den tekniske løsningen for uRO. Rapporten inneholder komponentene vi brukte for å bygge løsningen, hvordan kretsen fungerer og forklaring av kode. Vi tar også for oss utfordringer vi støtte på under arbeidet med tekniske løsninger, og hvordan vi løste disse. Rapporten avsluttes med et avsnitt om hvordan fremtiden kunne sett ut for loggføreren.

1. Om oss

Gruppen vår uRO, består av Lotte Stubdal (22), Kristine Løkke (20), Anna Helgestad (20), Ane Marthe Ness (20) og Renate Farberg (21). Vi er alle bachelorstudenter ved informatikk: design, bruk og interaksjon. uRO har slagordet - "Vi finner roen i uroen", og målgruppen vi har jobbet med er natteravnene.

2. Målet med loggføreren

Problemstillingen til dette prosjektet er:

"Hvordan kan vi hjelpe natteravnene med å utnytte ressursene sine på best mulig måte?"

Natteravnene har et problem med upraktisk og tungvint loggføring av hendelser, som de sender til politiet. For å svare på problemstillingen har vi derfor valgt å lage en løsning som gjør dette lettere. Loggføreren gjør det mulig å dokumentere hendelser ved hjelp av et gps-shield. I tillegg har den hendelserknapper som er delt opp i ulike type kategorier, og en opptaker som lagres på et SD-kort. På denne måten vil loggføring som forekommer i løpet av kvelden og natten gå mye raskere og være mer pålitelig i form av at alle hendelser blir logget med tid, dato og sted.

3. Presentasjon av prosjektvideo

Prosjektvideoen starter med å presentere flere scenarier som viser hvordan natteravnene pleier å loggføre, samt noen av utfordringene vi har kartlagt gjennom prosjektet. Ved hjelp av humoristiske hjelpemidler ønsker vi å vise hvordan natteravnene møter problemer når de loggfører ved hjelp av penn og papir. Midtveis i prosjektvideoen presenteres loggføreren for




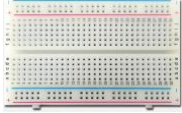

natteravnene som en reddende engel under knotete papirarbeid. I denne delen viser vi prototypens egenskaper og funksjoner. Til slutt vises noen tidligere scenarioer, men nå med bruk av loggføreren. Målet for prosjektvideoen er å illustrere hvordan natteravnene arbeider, og ikke minst bruken av loggføreren.





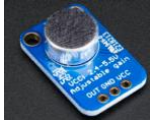

Link til video:

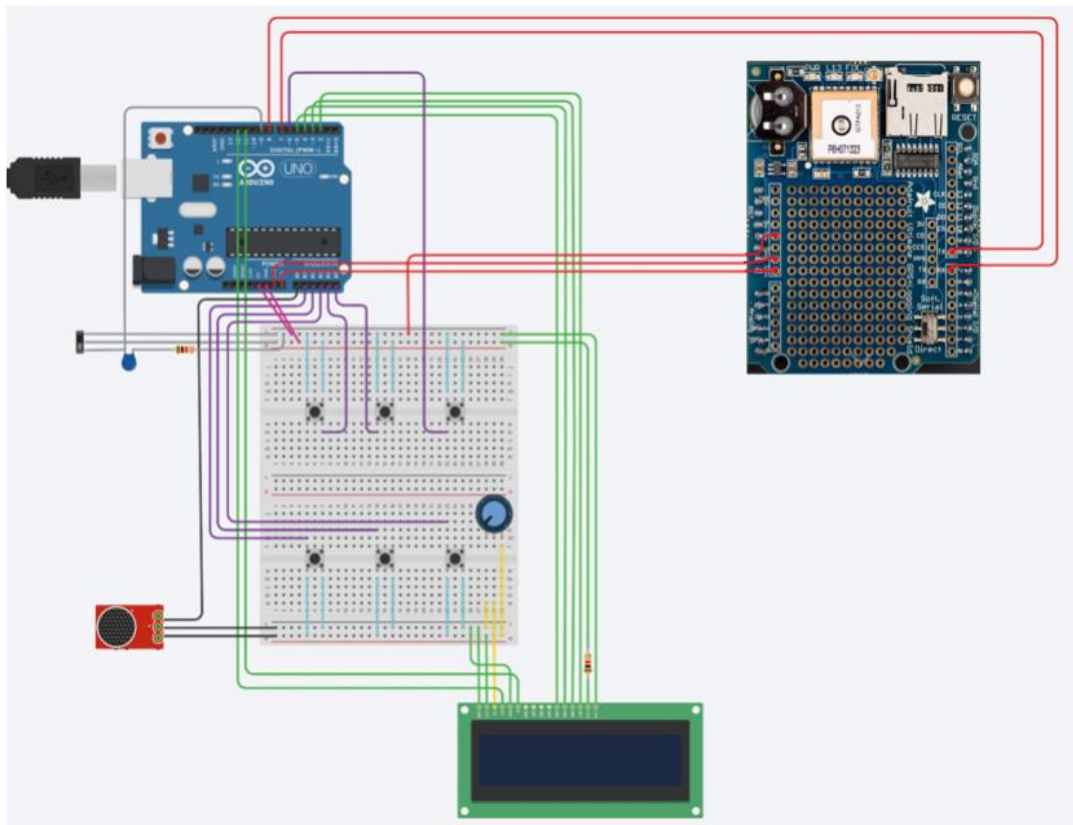
https://www.youtube.com/watch?v=IXwX_h6bBiA&fbclid=IwAR3XYUNmaEvmukp4WL8GeRbpxrvKtS4V-IGUKZr3pwBdoHWuqRWhgW9Uq0

4. Komponenter

I tabellen under viser vi komponentene vi har brukt for å lage loggføreren. Vi har skilt komponentene i forhold til hvor vi har kjøpt dem.

Komponent	Antall	Bilde
Fra Arduino Uno starterpack		
Arduino Uno	1	
Push button	6	
Ledninger	16	
Breadboard	2	
Resistor	8	

LCD skjerm	1	
Tilleggskomponenter fra Digital impuls		
Adafruit ultimate GPS logging shield	1	
Female - to - male wire	1 pk	
Tilt-switch	1	
Adafruit electret microphone amplifier	1	
Ekstra		
Power bank	1	



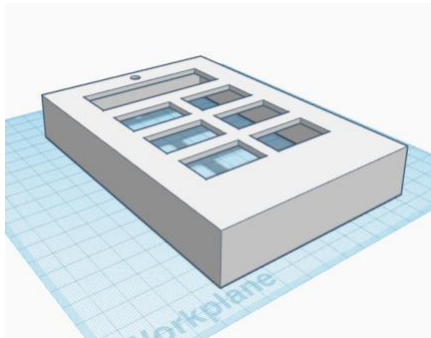
Illustrasjonen viser kretstegning av løsningen. Laget med <https://www.tinkercad.com/>

4.1 Bygging

Bildet over illustrerer kretsen til loggføreren. Et av kravene til loggføreren var at den skulle kunne henge rundt halsen slik at den var lett tilgjengelig under vandringene. For å nå dette målet ble vi nødt til å jobbe med plassbesparende tiltak. Det var også viktig for oss at størrelsen ikke skulle gå utover funksjonaliteten. For å få strøm til alle komponenter var vi nødt til å sette sammen to breadboard.

Et annet tiltak for plassbesparing var å kun bruke en arduino uno, istedenfor to. I utgangspunktet var det ikke nok digitale porter for alle komponentene våre, derfor var vi også nødt til benytte oss av de analoge portene. Vi brukte de analoge portene for knappene og vi loddet GPS-shieldet. GPS-shieldet er egentlig laget for å kunne settes rett i en arduino uno, men ved å lodde portene kunne vi benytte oss av male-female ledninger til portene i arduino uno. Mikrofonen ble også loddet slik at den kunne settes til åpningen på toppen av loggføreren.

For å gjøre løsningen så lett og minimalistisk som mulig valgte vi å 3D-printe loggføreren. Vi lagt inn hull til gps-antennen på baksiden av loggføreren, slik at den lett kunne fange inn signaler. Vi har også lagt inn hull på toppen slik at mikrofonen lettere fange opp lyd fra brukeren.



Illustrasjonen over viser loggføreren før den ble 3D-printet

4.2 utfordringer med komponentene

Underveis i prosjektet møtte vi på noen problemer med den tekniske løsningen. Vi begynte med å teste alle komponentene hver for seg, på denne måten kunne vi se om noen av dem hadde svakheter. De fleste komponentene fungerte godt, men vi fant fort problemer med mikrofonen. Vi kjøpte først en mikrofon som ikke klarte å ta opp noe annet enn svært skarpe og høye lyder. Etter dette søkte vi hjelp på Sonen hvor vi fikk låne en Adafruit microphone amplifier. Denne mikrofonen hadde også problemer med å plukke opp lyden, men ved å justere skruen bak fikk den tatt opp lavere lydfrekvenser. Når det kom til koding av mikrofonen støtte vi også på noen problemer.

```
void sound(){
  int sum = 0;
  for(int i=0; i<100; i++) {
    sum += analogRead(soundPin);
  }
  sum = sum/100;

  if(sum > 924){
    Serial.println("Sound ---- OK");
    writeSound();
  }
}
```

```
}  
  Serial.println("Sum " + sum);  
barrier  
  delay(10);  
}
```

Som vist i kodeutsnittet over er det satt en barriere for når den skal fange opp lyd. Denne barrieren var i noen tester for høy, og andre ganger for lav. For å løse dette problemet brukte vi mye tid på testing og finjustering, for å til slutt lande på den mest ideelle barrieren for natteravnenes bruk.

En annen utfordringen var den generelle strømflyten i kretsen. Loggføreren har mange komponenter og vi brukte derfor mye tid på å forstå hvor det ble for mye, eller for lite strøm. Grunnet uferdig krets fikk vi feilkoder som *“program is not responding”* og *“problem uploading to board”* gang på gang. Den eneste løsningen vi fant var å fjerne led-lyset fra løsningen. Tanken var originalt å ha led-lyset som en feedback til brukeren. På denne måten kunne man enkelt få bekreftelse på at lyden faktisk ble registrert. Lyset hadde spilt på designprinsippene affordance, consistency og feedback, men grunnet knapphet på tid fikk vi ikke laget en løsning med denne. Likevel var ikke dette avgjørende for hovedmålet til løsningen da det kun var en tilleggsfunksjon.

5. Kode

Koden til loggføreren er satt sammen av en gps-prekode, kode for en lcd-skjerm , knappene og mikrofonen.

Link til kode: <https://github.uio.no/renatfa/IN1060>

5.1 Forklaring av kode

I setup() blir knappene, lcd-skjermen, filen som gpsen skal skrive til og mikrofonen satt. Tilt-switchen er også en viktig del av setup() fordi det er den som bestemmer om kretsen får strøm eller ikke. Tilt-switchen fungerer som en tastelås. Når natteravnene er ferdig med å loggføre en hendelse kan de trykke på denne for å skru av loggføreren og unngå komme borti knappene ved uhell. Serial.begin(115200) blir koblet til denne verdien slik at gpsen blir lest

av raskt nok. `GPS.begin(9600)` og `GPS.sendCommand()` starter bruken av `gps-shieldet`. `GPS`-koden blir forklart i avsnitt 5.2.

I `loop()` leses knappene, som videre kaller på metodene som skal utføres. Knappene skal kontinuerlig sjekkes om de er trykket på eller ikke, samt om angreknappen har blitt trykket rett etter hendelsesknappen. I `loop()` har vi også *`int correctTime`* som bruker `millis()`. Denne brukes for å sjekke om brukeren har trykket på angre to sekunder etter en av de andre knappene. Koden er satt sammen av mange hjelpemetoder som utføres ved knappetrykk, hvorav alle inneholder `delay`. Dette er fordi det ikke er nødvendig å kunne interagere med systemet de tre sekundene det skrives ut på `lcd`-skjermen.

Hvis for eksempel `button3` blir trykket på, vil det først sjekkes om `button6(angre)` er trykket på, for å deretter kalle på `void drunkPerson()` eller `undo()`. If-sjekken:

```
if(pushes(button6) || count > 1){
    undo();
}
```

kaller på metoden `pushes()` med angreknappen som parameter. `Pushes()` er en metode som ved hjelp av `millis` sjekker om angreknappen er trykket innefor to sekundene. Vi valgte å ha to sekunder som tidsfrist fordi vi ønsker at angreknappen skal fungere hvis brukeren trykker på feil knapp, ikke for å kunne fjerne en hendelse fra `sd`-kortet senere. Metoden har en teller, ”`count`”, som blir oppdatert. I kodeutsnittet over viser vi hvordan metoden kaller på `undo()` hvis den returnerer positivt. `Undo()` er kun en metode som gir feedback til bruker om at hendelsen ikke ble loggført. Hvis if-sjekken (vist i kodeutsnittet) returnerer positivt skal hendelsen loggføres. I metoden `drunkPerson()` settes først `lcd`-skjermen til å bli tom, deretter skrives feedback ut i tre sekunder. Metoden kaller også på `GPSPMethod()` og `writeToLog()`, som forklares i avsnitt 5.2. Til slutt kalles `homeScreen()`, som skriver ut riktig klokkeslett og dato på skjermen.

Metoden `void sound()` blir kalt på hvis brukeren vil ta opptak. Metoden leser av summen av mikrofonen, og sjekker om lydverdien er høy nok til at den skal tas opp. Hvis lyden er under barrieren vi har satt kategoriseres det som bakgrunnsstøy, og det skal derfor ikke bli ”godkjent” som et lydopptak. Når if-sjekken slår til kaller den på `writeSound()`, som gir feedback til brukeren. Lydbarrieren er satt på grunnlag av testing og justering av mikrofonen.

5.2 GPS shield

GPS shieldet er koblet til 5V, GND, Vin, D8 og D7. Vi valgte å bruke en prekode fra Adaruit biblioteket, fordi den inneholdt alle funksjonene vi trengte for å ta opp signaler og lagre de på sd-kort. Så lenge shieldet får strøm vil den lete etter fix, og når den har det vil den skrive til en fil på sd-kortet. Fix er tiden som trengs for å få kontakt med nok satellitter for å kunne finne koordinatene til loggføreren. Vi har valgt å forenkle prekoden ved å sette koden inn i mindre metoder slik at den blir mer oversiktlig. GPSMethod() sjekker først om shieldet er avbrutt, hvis det ikke er det kaller koden på GPS.read(). Når gps-en leses oppdateres klokken, sted og dato som skrives ut i terminalen. Dette gjorde det enklere å teste om gps fix-en var god nok.

Metoden writeToLog() kalles på i hver av hendelsesmetodene. Metoden skriver inn gps-verdiene hvis den får fix. Hvis den er inne eller i trange bakgater vil den mest sannsynlig ikke kunne klare å fullføre writeToLog(), fordi her vil det være vanskelig å få signaler.

Metodene getTime() og getDate() leser av verdien fra GPS.read() for å kunne skrive ut riktig verdi på lcd-skjermen.

6. Hvordan har kravene blitt nådd?

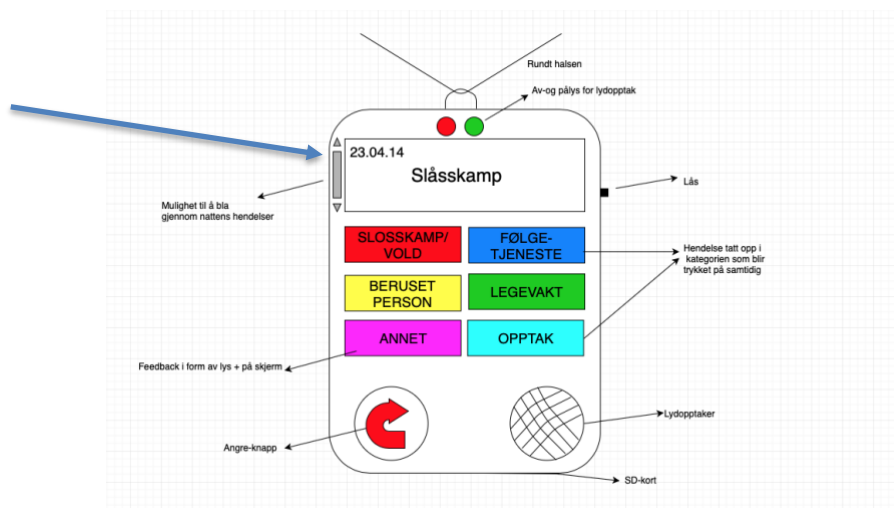
Funksjonelle krav	Ikke-funksjonelle krav	Hvordan målene blir nådd
Prototypen skal kunne brukes på alle årstider	Knappene skal kunne trykkes på med votter	Størrelsen på knappene gjør de enkle å trykke på. En tastelås er implementert slik at det unngås feiltrykk.
Prototypen skal loggføre raskt	Prototypen skal loggføre innen 1 sekund hvis den har fix	Lcd-skjermen gir feedback i form av utskrift i 3 sek etter hendelsen er loggført.
Prototypen skal lagre relevant data	Prototypen skal lagre nøyaktig sted, tid og dato for hendelse	GPS.read() og skrivTilLogg() legger inn koordinater, tid og dato i en fil på sd-kort

Prototypen skal være enkel å ha med seg på vandring	Prototypen skal kunne bæres rundt halsen med ytterklær på	Valg av lett materiale (plast)
Prototypen skal være robust	Prototypen skal tåle fall fra 2 m	Valg av hardplast som materiale

7. Veien videre

Hvis vi hadde hatt lengre tid på prosjektet ville vi lagt til og forbedret loggføreren ytterligere. Gruppen hadde ønske og mål om å få til flere funksjoner ved løsningen, men grunnet tidsbegrensning har dette ikke blitt en realitet. Vi ville først og fremst fått lampen til å fungere med resten av kretsen, fordi vi tror dette ville gjort løsningen mer intuitiv. Vi hadde også en ambisjon om at knappene skulle kunne fungere sammen. Ved å for eksempel kunne trykke på slåsskamp og deretter lydopptak kunne man lettere kategorisere lydfilene. Dette ville gjort det lettere å gå gjennom dataen etter en vandring, som dermed ville nådd målet om å effektivisere natteravnens arbeid ytterligere. Loggføreren tar på dette tidspunktet opp lyd, men den blir derimot ikke lagret noe sted. Hvis vi hadde hatt bedre tid ville vi lagret lydfilene på et eget sd-kort slik at organiseringen hadde blitt oversiktlig.

En annen funksjon vi har snakket om er å kunne ha et hjul som natteravnene kunne brukt for å se på hendelsene som ble loggført den kvelden. Hjulet hadde gjort det lett å navigere seg gjennom en liste fra sd-kortet, som vil vises på skjermen. Vi tror denne funksjonen hadde vært nyttig når natteravnene møtes underveis i vandringene.



Illustrasjonen vi først lagde av loggføreren. Bildet viser tanken bak hjulet

En annen funksjon som hadde vært nyttig for bruken av loggføreren er hvis den hadde vist et symbol på lcd-skjermen når den har fix. På denne måten hadde brukeren enkelt kunne visst når det trengtes bedre signalforhold. Dette er noe vi kom på etter vi hadde ferdigstilt og brukertestet prototypen. Denne funksjonen hadde vi lett kunne implementert ved for eksempel:

```
void fixSignal(){
  lcd.setCursor(0,0);
  if(GPS.fix){
    lcd.print("*");
    lcd.setCursor(5,0);
  }
}
```

I en ideell verden hadde loggføreren fungert på telenettverket, slik at den kunne kommunisert mellom vandringsgruppene. Dersom flere vandringsgrupper er ute å går ville løsningen da oppdatert alle vandringsgruppene hvis noen hadde loggført noe. Hvis løsningen hadde støttet telenettverket ville man også kunne sendt all loggført data trådløst.

8. Kildeliste

Lady ada(Jun 08, 2019): Adafruit Ultimate GPS Logger Shield. 01.05.2019. Henet fra <https://learn.adafruit.com/adafruit-ultimate-gps-logger-shield> >