

Gruppe 27: Eureka!

Teknisk rapport

Emnekode: IN1060

Gruppemedlemmer: Mikael Sørli Niane, Kjersti Ås Røberg, Subinur Mahsut, Oliver Thomas Kusiak, Synnøve Berg Vålerhaugen og Pia Larsen.

Antall ord: 3159

Dato: 31. mai 2023



Innholdsfortegnelse

Prosjekt: "Søvnhygiene"	1
Funksjoner.....	1
Video	2
Storyboard.....	2
Del 1: Bruksscenario - 3 morgener.....	2
Del 2: Demo av artefakt - testing.....	3
Om implementasjonen.....	3
Videolenke.....	4
Komponenter brukt	5
Teknisk løsning - kode.....	7
GitHub-lenke.....	7
Temperatur- og luftfuktighet.....	7
Tid - klokke modul.....	8
Dimmefunksjon.....	9
Hovedprogram.....	11
Utfordringer.....	17
Referanser	19

Prosjekt: “Søvnhygiene”

Vi valgte temaet “Sanse det ikke-sansbare”. Målet for prosjektet er å utvikle løsninger som kan hjelpe til å etablere en god søvnhygiene for å bidra til å forbedre søvn, ved å hjelpe brukere til å finne ut hvilke faktorer som forstyrrer søvn og til å danne en fast døgnrytme. Sensorteknologi gjør det mulig å måle og registrere fenomener rundt oss som vi ikke er sansbare eller som er vanskelig å påvirke for mennesker i det daglige (Bratteteig, 2021, s.124-125). Derfor har vi laget en prototype som benytter flere ulike sensorer som vil kunne måle og gi nyttig informasjon til brukerne. Vi har gitt prototypen vår navnet “SleepWell” som er inspirert av vårt mål med prosjektet.

Funksjoner

Ved bruk av en temperatur- og luftfuktighetssensor så får vi sjekket hvordan temperaturen er i rommet, der varsellampen vil lyse dersom den er under eller over en viss temperatur (15-22 grader). På samme måte bruker vi temperatur- og luftfuktighetssensoren for å måle luftfuktigheten i rommet, der lampen skrur på hvis luftfuktigheten er under 20% eller over 60%. Dette er faktorer som kan forstyrre søvn (Svartdal, 2023; Temperatur, fukt og trekk er viktig for kroppens varmebalanse, 2012), og vil kunne være nyttig for brukerne å få tilbakemelding på. Måleren logger hele døgnet, men gir tilbakemelding kun på kveld- og nattmålingene ettersom dette er målet med prototypen og prosjektet vårt. Men det kan være nyttig å ha dataene fra dagtid.

Videre har vi brukt en klokke modul slik at artefakten vet hva klokken er, noe som er vesentlig hvis artefakten skal brukes som hjelp for oppvåkning i form av lys - noe som var ønskelig hos flere av våre brukere. Ved oppvåkning har vi brukt en lypsære fra en lommelykt som lyskilde. Denne blir dimmet opp gradvis over en periode på 30 minutter før oppvåkningstiden. Dette vil kunne bidra til en fast søvnrutine ved å gi en rolig oppvåkning og et fast oppvåkningstidspunkt (Svartdal, 2023). Oppvåkningstiden blir satt av bruker ved å holde inne temperaturknappen i 5 sekunder. For å gi tilbakemelding på at tidspunkt for oppvåkningstiden er satt blinker de to røde LEDene tre ganger.

Video

Videoen tar både for seg prototypen i brukskonteksten, og viser en demo av hvordan prototypen fungerer teknisk.

Storyboard



Del 1: Bruksscenario - 3 morgener

1. morgen: Bruker våkner opp kl 7 fra bråkete alarm, trøtt og sliten med hjertebank. Går i butikken og kjøper en SleepWell. På kvelden før bruker skal legge seg lyser varsellampene for temperatur og luftfuktighet. Hun åpner vinduet, og skruer av varmeovnen, før hun trykker på henholdsvis temperaturknapp og luftfuktighetsknapp og legger seg.
2. morgen: Bruker våkner opp av lampen som lyser kl 7. Varsellampene for temperatur og luftfuktighet lyser rødt. Bruker skruer disse av ved å trykke på knappene. Hun setter

oppvåkningstidspunkt på SleepWell ved å holde inne temperaturknappen i 5 sekunder og den blinker tre ganger.

3. morgen: Kl 06.30 begynner dimmingen av lyset. Bruker våkner opp av lampen kl 7, blid og uthvilt. Ingen varsellamper lyser.

Del 2: Demo av artefakt - testing

For å demonstrere hvordan artefakten fungerer blåses varm fuktig luft mot artefaktens temperatur- og luftfuktighetssensor. Dette gir høye målinger utenfor referanseområdet. Her i demonstrasjonen er øvre temperaturgrense satt til 24 grader ettersom det er filmet på dagtid i en stue i et vanlig hjem, og det vanskelig å kontrollere den høye temperaturen på dagtid for å komme innenfor referanseområdet når man skrur av varsellampene.

Når temperaturen går ned (visning av skjerm med Arduinokode som printer ut nåværende temperatur og luftfuktighet) trykkes knappene for luftfuktighet og temperatur, og varsellampene skrur av. Deretter skrur oppvåkingslampen av ved at det trykkes på kuppelen.

Om implementasjonen

Dimmefunksjonen er ikke en del av implementasjonen (se utfordringer), men er demonstrert i videoen ved at vi bruker lyset fra den hvite LEDen for å gi et inntrykk av lysets funksjon i brukskonteksten. Dimmefunksjonen er derimot implementert i koden.

Ettersom lagring av data fra målingene ikke er mulig i dette prosjektet ut ifra begrensningene som er satt (se utfordringer) er dette ikke en del av implementasjonen og heller ikke tatt med i videoen, men en tenkt mulighet for det endelige produktet.

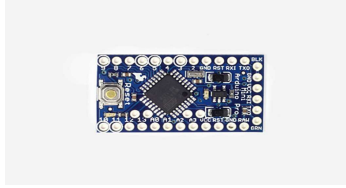
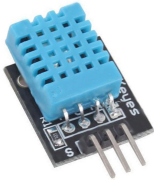
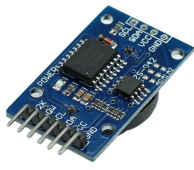


Det er også slik at storyboardet som er vedlagt ovenfor beskriver en mer omfattende case, og vi har dermed ikke en like beskrivende video.






Videolenke

https://www.youtube.com/watch?v=VOYk_TYIeLs

Teknisk rapport

Komponenter brukt

Komponent	Benevnelse	Funksjon	Bilde/bilder av komponentet
Arduino Pro Mini	ATMEGA328P	Utviklingskort	 Figur 1
Temperature and Humidity Sensor Module	DHT11	Måler temperatur og luftfuktighet	 Figur 2
Real Time Clock RTC Module	DS3231	Setter et klokkeslett og dato, og opprettholder disse	 Figur 3
3V Batteri	CR2032	Alternativ strømkilde til klokkemodulen	 Figur 4
Knapp temperatur og lys(sette tidspunkt)		<ul style="list-style-type: none">- Trykkes for å skru av rød LED temperatur- Holdes inne i 5 sekunder for å sette nytt tidspunkt for oppvåkning/lys	 Figur 5. Bilde av pushbutton, Oslo 28. mai 2023

Knapp luftfuktighet		Trykkes for å skru av rød LED luftfuktighet	 <p>Figur 5. Bilde av pushbutton, Oslo 28. mai 2023</p>
Knapp lampe		Trykkes for å skru av lampen	 <p>Figur 5. Bilde av pushbutton, Oslo 28. mai 2023</p>
Rød LED temperatur		Lyser ved for høy/lav temperatur	 <p>Figur 6. Bilde av rød LED, Oslo 28. mai 2023</p>
Rød LED luftfuktighet		Lyser ved for høy/lav luftfuktighet	 <p>Figur 6. Bilde av rød LED, Oslo 28. mai 2023</p>
Hvit LED		Lyser gradvis opp 30 minutter før angitt oppvåkningstidspunkt	 <p>Figur 7</p>

Teknisk løsning - kode

GitHub-lenke

<https://github.uio.no/pialar/IN1060prosjekt>

Temperatur- og luftfuktighet

Vi begynte arbeidet med programmeringen av prototypen med å finne dokumentasjon for temperatur- og luftfuktighetssensoren, og teste ut koden med oppsett med sensor på Arduino.

```
#include <dht11.h>
#define DHT11PIN 4

dht11 DHT11;

void setup(){
  Serial.begin(9600);
}

void loop(){
  Serial.println();

  int chk = DHT11.read(DHT11PIN);

  Serial.print("Humidity (%): ");
  Serial.println((float)DHT11.humidity, 2);

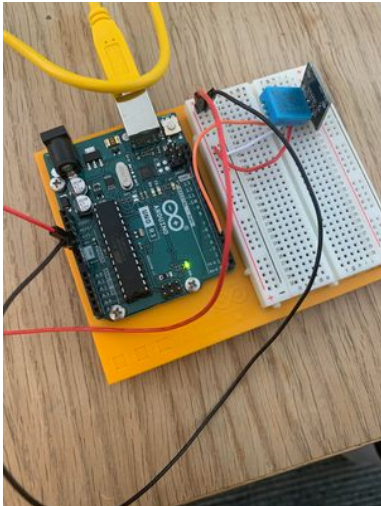
  Serial.print("Temperature (C): ");
  Serial.println((float)DHT11.temperature, 2);

  delay(2000);
}
```

(TempOgFuktSensor.ino)

Koden leser pin til temperatur- og luftfuktighetssensoren, og printer målingene av luftfuktighet og temperatur med et opphold på 2 sekunder.

I Hovedprogrammet byttet vi til DHT.h-biblioteket av Adafruit da dette var enklere å bruke.



Testoppsett av temperatur- og luftfuktighetssensor.

Tid - klokke modul

Deretter ville vi ha en funksjon for at artefakten skal kunne stille inn tidspunkt, og valgte derfor å bruke en klokke modul for å holde tiden. Koden setter et klokkeslett og dato, og klokken opprettholder dette ved hjelp av enten VCC eller batteriet (dersom ikke VCC er tilgjengelig). Artefakten vil fremdeles gå på strøm (via ledning til strømuttak), men ettersom modulen har batteri vil den kunne fortsette å holde tiden dersom artefakten kobles fra strømuttaket.

```
#include <Wire.h>
#include <RtcDS3231.h> //Rtc by Makuna

//Oppretter nytt RTC objekt
RtcDS3231<TwoWire> rtcModule(Wire);

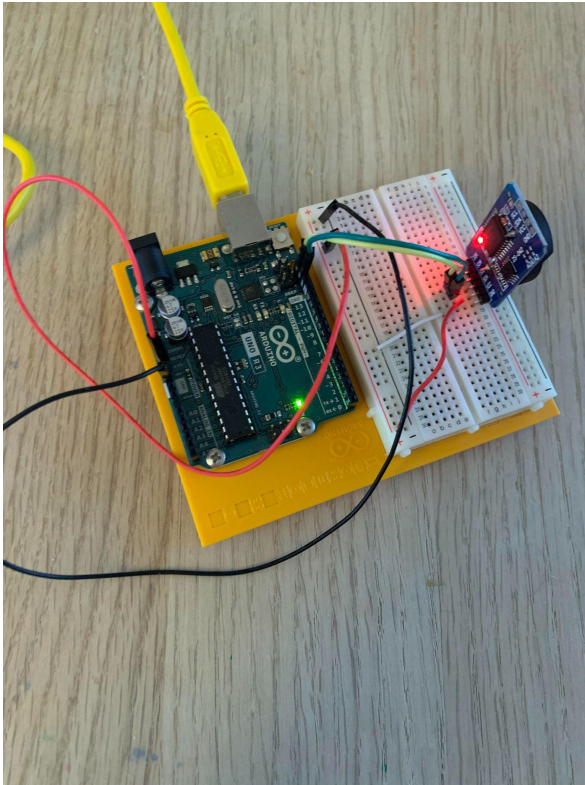
void setup() {
  Wire.begin();

  RtcDateTime compiled = RtcDateTime(__DATE__, __TIME__);
  rtcModule.SetDateTime(compiled);
}

void loop(void) {
}
```

(SettTid.ino)

Koden henter Wire.h-biblioteket, og lar brukeren fylle inn et starttidspunkt ("initial time"). Serial.print printer nåværende tidspunkt hvert sekund.



Testoppsett av klokke modul.

Dimmefunksjon

Dimmefunksjonen er en funksjon som tar imot tre parametere - nåværende tid, ønsket oppvåkningstid og en boolean verdi for om "skru av" knappen er trykket. Funksjonen sin hovedoppgave er å dimme lyset opp fra null til max lysstyrke over de siste 30 minuttene før man ønsker å stå opp. Det funksjonen da gjør er å først sjekke om "skru av" knappen er trykket, og hvis den er det så vil lyset skrues av eller forbli avskrudd. Deretter, hvis knappen ikke er trykket, sjekker funksjonen nåværende tid opp mot oppvåkningstiden for å se hvor lenge det er til. Dersom det er under 30 minutter til oppvåkning vil lyset bli dimmet etter hvor lenge det er igjen. Her brukes Arduino sin PWM for å "dimme" lyset - dette er egentlig bare at det blinker veldig fort av og på slik at det får en effekt lik et dimmet lys. Lysstyrken settes på en skala fra 0 til 255 etter hvor lenge det er igjen, dette på formelen $255 (\text{max}) - \text{gjenstående tid} * 8,5$. 8,5 er 1/30 del av 255, som vil si at ved 30 min igjen så har den lysstyrken 0, ved 15 min igjen har den lysstyrken 127,5 og ved 0 minutter igjen har den lysstyrken 255. Fra 0 minutter igjen og fremover i tid (etter oppvåkningstiden) vil lyset lyse på full styrke frem til "skru av" knappen trykkes.

```

void dimmerFunksjon(RtcDateTime naaTid, RtcDateTime staaOppTid, bool lysKnappTrykket) {

int minutesUntilWakeUp = 31;

// Hvis knappen er trykket skal lyset være av
// Denne nullstrilles i hovedprogramet?
if (lysKnappTrykket) {
    analogWrite(ledPin, LOW);
    return;
}

if (naaTid.Hour() == staaOppTid.Hour()) {
    minutesUntilWakeUp = staaOppTid.Minute() - naaTid.Minute();
} else if ((staaOppTid.Hour() - naaTid.Hour()) == 1 ) {
    minutesUntilWakeUp = (60 - naaTid.Minute()) + staaOppTid.Minute();
}

// La lyset stå på fullt til det blir skrudd av med knapp
if (minutesUntilWakeUp <= 0) {
    analogWrite(ledPin, HIGH);
} else if (minutesUntilWakeUp <= 30) {
    // Dim lyset fra verdi 0 til 255 over 30 min
    int lysstyrke = 255 - (minutesUntilWakeUp * 8.5);
    analogWrite(ledPin, lysstyrke);
}
}

```

Hovedprogram

Hovedprogrammet importerer Wire.h- biblioteket for kommunikasjon med SCL- og SDA-pinene på klokkemodulen, LysDimmer.h for dimming av lys, RtcDS3231.h-biblioteket for klokkemodulen, og DHT.h.

Vi definerer type og pin til DHT-biblioteket.

Deretter defineres alle pinene brukt til sensor, knapper og LEDer.

I linje 16-18 opprettes instanser av de tre klassene.

Fra linje 20 til 27 deklarerer vi alle variabler.

I setup() brukes Serial.begin, og pinMode settes for alle knapper og LEDer.

dht.begin()-funksjonen hentes. Så initieres variabelen internTid med funksjonen millis() for lampen og dens funksjoner.

void loop() sjekker om fuktKnapp er trykket, og setter state på fukt-LEDen til LOW. void loop kaller også på funksjonen tempKnappTrykket(), som sjekker om knappen holdes inne for å endre tidspunkt på når brukeren vil våkne/vekkes av lyset. lysKnappTrykkes settes til true.

Funksjonen tempKnappTrykket kaller igjen på funksjonen tempKnappSession. Denne sjekker om knappen har blitt sluppet 50 ganger i løpet av 5 sekunder. Hvis den ikke har blitt sluppet oppdateres stå opp-tiden. Dersom knappen slippes skrur funksjonen tempKnappSession av temperatur-LEDen. Når funksjonen da kalles i funksjonen tempKnappTrykket lagres returverdien i variabelen tempKnappKommando. Dersom denne er "NY_TID" kalles nåværende klokkeslett fra klokkemodulen gjennom funksjonen GetDateTime. Vi bruker digitalWrite() og delay til å få tempLed og fuktLed til å blinke tre ganger av og på, og staaOppTid oppdateres. Hvis returverdien er "RESET_TEMP" settes tempLedState til LOW.

Deretter kalles sjekkTempogFuktSensor for å lese av sensoren og oppdaterer lampen en gang per sekund.

Funksjonen sjekkTempOgFuktSensor kaller på funksjonene dht.readTemperature() og dht.readHumidty, og lagrer dem i variabler, som gjennom if-test sammenliknes med

ønskede/anbefalte verdier for temperatur og luftfuktighet, og setter state til HIGH dersom det er utenfor referanseområdet.

Tiden hentes fra klokkemodulen og legges i variabelen naa. Så kalles funksjonen resetLysTrykktState for å sjekke om det er mer enn én time siden oppvåkningstiden for å skru av lampen. Den bruker funksjonene .Hour() og .Minute() for å sjekke hvor langt ut på dagen oppvåkningstiden er sammenliknet med nåværende tidspunkt, og sjekker om det er én time eller mer siden oppvåkningstiden. I så fall settes lysKnappTrykkt til false.

Deretter kalles dimmefunksjonen fra LysDimmer.h-biblioteket som gradvis setter opp lyset på lampen (se Dimmefunksjon over). Ved hjelp av digitalWrite() oppdateres temp- og fukt-LEDene med nye statuser, deretter oppdateres interntid (med millis()).

(hovedprogram.ino)

```
#include<Wire.h>
#include<RtcDS3231.h>
#include<LysDimmer.h>
#include<DHT.h>

#define DHTTYPE DHT11
#define DHTPIN 4

int lysknapp = 5;
int tempKnapp = 6;
int fuktKnapp = 7;
int tempLed = 8;
int fuktLed = 9;
int lysPin = 10;

RtcDS3231<TwoWire> klokkeModul(Wire);
LysDimmer lysDimmer(lysPin);
DHT dht(DHTPIN, DHTTYPE);

RtcDateTime naa;
RtcDateTime staaOppTid;
unsigned long internTid;
bool tempLedState = LOW;
bool fuktLedState = LOW;
bool lysknappTrykkt = false;
float tempVerdi, fuktVerdi;
String tempKnappKommando;
```

```

void setup() {
  Serial.begin(9600);
  pinMode(lysknapp, INPUT);
  pinMode(tempKnapp, INPUT);
  pinMode(fuktKnapp, INPUT);
  pinMode(tempLed, OUTPUT);
  pinMode(fuktLed, OUTPUT);
  dht.begin();

  //Initierer interntid;
  internTid = millis();
}

void loop() {

  if (digitalRead(tempKnapp)){
    tempKnappTrykktet();
  }

  if (digitalRead(fuktKnapp)){
    fuktLedState = LOW;
  }

  if (digitalRead(lysknapp)){
    lysknappTrykktet = true;
  }

  //Leser sensorer og oppdaterer lampen maks en gang pr sekund.
  if (millis() - internTid > 1000){
    sjekkTempOgFuktSensor();

    //Henter tiden fra klokkemodulen
    naa = klokke modul.GetDateTime();

    //Én time etter oppvåkningstid resettes lysKnappTrykktet
    if (lysknappTrykktet){
      resetLysTrykktetState(naa);
    }

    //kaller dimmerfunksjon som styrer dimmer
    lysDimmer.dimmerFunksjon(naa, staaOppTid, lysknappTrykktet);

    //Oppdaterer temp og fukt-led med nye statuser
    digitalWrite(tempLed, tempLedState);
  }
}

```

```

digitalWrite(fuktLed, fuktLedState);

//Oppdaterer interntid
internTid = millis();
}
}

void tempKnappTrykket() {
//Hvis tempknapp holdes inne, vil alarmen settes til det tidspunktet
tempKnappKommando = tempKnappSession();
if ( tempKnappKommando == "NY_TID"){
    staaOppTid = klokkemodul.GetDateTime();
    digitalWrite(tempLed, HIGH);
    digitalWrite(fuktLed, HIGH);
    delay(400);
    digitalWrite(tempLed, LOW);
    digitalWrite(fuktLed, LOW);
    delay(400);
    digitalWrite(tempLed, HIGH);
    digitalWrite(fuktLed, HIGH);
    delay(400);
    digitalWrite(tempLed, LOW);
    digitalWrite(fuktLed, LOW);
    delay(400);
    digitalWrite(tempLed, HIGH);
    digitalWrite(fuktLed, HIGH);
    delay(400);
    digitalWrite(tempLed, LOW);
    digitalWrite(fuktLed, LOW);
}
else if (tempKnappKommando == "RESET_TEMP"){
    tempLedState = LOW;
}
}

String tempKnappSession() {
//Sjekker om knapp har blitt sluppet 50 ganger på 5 sekunder. Hvis ikke så oppdateres 'stå opp'-tid
for (int i = 0; i <= 50; i++){
    if (!digitalRead(tempKnapp)){
        return "RESET_TEMP";
    }
    delay(100);
}
return "NY_TID";
}
}

```

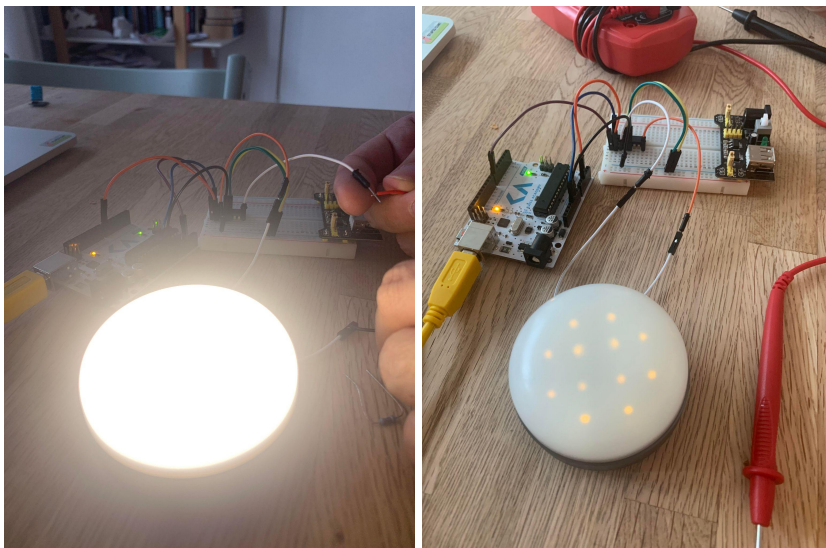


```

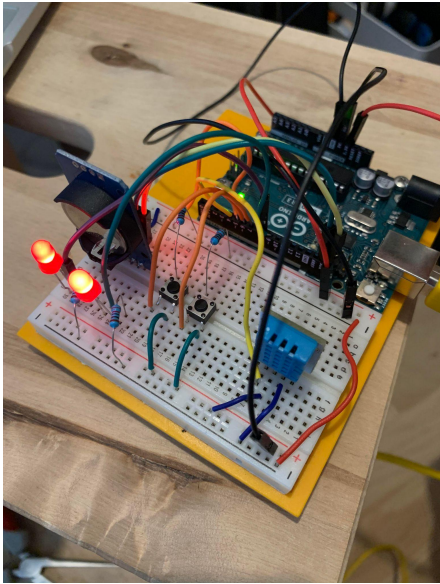
void resetLysTrykktState(RtcDateTime datoTidNaa){
int staaOppTidIMin = staaOppTid.Hour() * 60 + staaOppTid.Minute();
int datoTidNaaIMin = datoTidNaa.Hour() * 60 + datoTidNaa.Minute();
if (datoTidNaaIMin - staaOppTidIMin >= 60){
  lysknappTrykkt = false;
}
}

void sjekkTempOgFuktSensor(){
tempVerdi = dht.readTemperature() - 2.4; //Kallibrering av aktuell sensor.
fuktVerdi = dht.readHumidity();
Serial.print("\n Temp: ");
Serial.print(tempVerdi);
Serial.print(" , Fukt: ");
Serial.print(fuktVerdi);
if (tempVerdi < 15.0 || tempVerdi > 22){ //Et spenn med temperaturverdier
  tempLedState = HIGH;
}
if (fuktVerdi < 20 || fuktVerdi > 60){ // Et spenn med godkjente luftfuktighetsprosent
  fuktLedState = HIGH;
}
}
}

```



Testing av dimming av hvit LED



Oppsett med temperatur- og luftfuktighetssensor, klokkemodul, røde LEDer og knapper.



Ferdig artefakt

Utfordringer

Under utviklingen av den tekniske løsningen ved prototypen møtte vi på flere utfordringer. En utfordring var brukernes ønske om å lagre dataen. Dette var ikke en mulig funksjon da vi ikke kan ta i bruk en app eller gjøre om dataen vi får til et dokument. Det ville derimot være enkelt å hente dataene, ettersom vi har dataene fra målingene og vi kunne sendt disse med `Serial.println()` til for eksempel en Raspberry Pi. Dette er derfor en teoretisk mulighet ved produktet vårt, men som ikke ble tatt med i utformingen av produktet grunnet begrensningene ved oppgaven i IN1060.

Et annet ønske brukerne hadde er en funksjon der brukeren selv kunne stille inn ønsket temperatur og luftfuktighet, stille inn vekketid selv, lengde på dimming og ønskede verdier på både temperatur og luftfuktighet. Disse er vanskelige å implementere uten en app eller skjerm. Vi løste det å kunne stille inn vekketid selv med en funksjon der man holder inne en knapp (temperaturknappen) i 5 sekunder på det tidspunktet dagen før man ønsker vekking. Vi la også til en funksjon der lyset på de to røde LEDene blinker tre ganger for å gi tydelig tilbakemelding på satt vekketid uten bruk av skjerm eller app.

Vi møtte også på en teknisk utfordring når det gjelder valg av referanseområde for temperatur. Brukerne nevnte 18-20 som en ideell temperatur på soverommet, men de var ikke sikre på om dette egentlig var ønsket temperatur eller bare hva de forestiller seg som en ideell temperatur. Ettersom vi ikke har noen teknisk løsning for å la brukerne sette sin egen ideelle temperatur så valgte vi derfor å ta utgangspunkt i nedre grense på 15 grader og vår øvre grense på 22 basert på anbefalinger fra flere kilder (Sjøgren, 2013; Råd for bedre søvn, 2022; Onen et al., 1994)

I prosessen med å utvikle prototypen støtte vi på utfordringer rundt størrelse på selve omslaget/boksen (laget av transparent og matt hvitt filament ved hjelp av 3D-printer) og hvordan vi kunne implementere de ulike komponentene i denne. Det var viktig for oss å oppfylle brukernes krav om et lite og hendig produkt. Vi brukte Arduino Pro Mini som er plasseffektiv, og lodding og plassering av komponentene er nøye gjennomtenkt. Det som skapte problemer for oss ved implementasjonen var stivheten til ledningene. Dessverre førte det til at den ene knappen (luftfuktighet) ikke fungerte på siste evaluering med brukerne, og en kortslutning hindret LEDen i å lyse. Dette rettet vi opp senere, men prosessen med å få

prototypen klar med alle funksjoner ved hjelp av det utstyret vi har tilgang på har vært krevende.

Ved testing av oppsett og kode for dimmefunksjon og hvit LED møtte vi på utfordringer rundt tilgang på komponenter som dimbare LEDer og dimbar LED-driver. Vi forsøkte med chipen L293D, men denne fungerer til PWM på DC-motorer og fungerte ikke godt med lys. Derfor er selve dimmingen ikke demonstrert med prototypen, men den hvite LEDen lyser og kan skrur av ved å trykke på kuppelen på prototypen.

Selv om det har vært mange utfordringer på veien og den tekniske delen av oppgaven med utvikling av prototypen har krevd mye arbeid, tid, gjennomtenkte valg, prøving og feiling, har vi kommet i mål med en høyoppløselig prototype som fungerer godt med de tiltenkte funksjonene, og konkretiserer brukernes ønsker og behov.

Referanser

- Bratteteig, T. (2021). *Design for, med, og av brukere*. Universitetsforlaget.
- Svartdal, F. (2023, 11. april). Søvnhygiene. I *Store norske leksikon*.
<https://snl.no/s%C3%B8vnhygiene>
- Folkehelseinstituttet (2015, 3.mai) Temperatur, fukt og trekk er viktig for kroppens varmebalanse.<https://www.fhi.no/ml/miljo/inneklima/artikler-inneklima-og-helseplager/temperatur-fukt-og-trekk-er-viktig/>
- Sjøgren, K. (2013, 22. januar). Hvilken temperatur er best å sove i?. I *Forskning.no*.
<https://forskning.no/sovn/hvilken-temperatur-er-best-a-sove-i/658961>
- Råd for bedre søvn. (2022, 22. juni). I *NHI.no*. Hentet 28. mai 2023 fra
<https://nhi.no/livsstil/egenomsorg/sovnhygieniske-rad/>
- Onen, S. H., Onen, F., Bailly, D. & Parquet, P. (1994, 12. mars). [Prevention and treatment of sleep disorders through regulation] of sleeping habits]. I *National Library of Medicine*. <https://pubmed.ncbi.nlm.nih.gov/8022726/>
- Kjell & Company. (2009-2016). *Hvordan virker Arduino?: Lær fra grunnen av og med raskt resultat* (Utgave 7.01). Ansvarlig utgiver: Karl Emil Nikka.
- arcaegecengiz. (2018, 23. november). *Using DHT11*. Project Hub.Arduino.
<https://projecthub.arduino.cc/arcaegecengiz/12f621d5-055f-41fe-965d-a596fcc594f6>
- (Forfatter: Ukjent). (2018, 28. januar). *A DHT11 Class for Arduino*. Playground.Arduino. <https://playground.arduino.cc/Main/DHT11Lib/>
- Adafruit Industries. (2023). *DHT-sensor-library*. GitHub.
<https://github.com/adafruit/DHT-sensor-library>

- Figur 1. Arduino Pro Mini. Fra DOCS.Arduino. (Forfatter: Ukjent) 16/5-2023.
(Nedlastningsdato: 28/5-23)
(<https://docs.arduino.cc/retired/boards/arduino-pro-mini>)

- Figur 2. DHT11–Temperature and Humidity Sensor. Fra Components101.
(Forfatter: Ukjent) 16/7-2021. (Nedlastningsdato 29/5-23)

(<https://components101.com/sensors/dht11-temperature-sensor>)

- Figur 3. DS3231 AT24C32 IIC Precision RTC sanntidsklokkehukommelsesmodul.
Fra Kuongshun Electronic Limited. (Forfatter: Ukjent). (Nedlastningsdato: 29/5-23)
(<http://no.sz-kuongshun.com/uno/uno-board-shield/ds3231-at24c32-iic-precision-rtc-real-time.html>)

- Figur 4. CR2032 Lithium Battery, 6 pack. Fra Biltema. (Forfatter: Ukjent)
(Nedlastningsdato: 28/5-23)
(<https://www.biltema.no/en-no/office---technology/batteries/cell-batteries/cr2032-lithium-battery-6-pack-2000044960>)

- Figur 7. Ledsavers Mini Campinglampe. Fra Kjell & Company.(Forfatter: Ukjent)
(Nedlastningsdato: 27/5-23)
(<https://www.kjell.com/no/produkter/hjem-fritid/belysning-lamper/arbeidsbelysning/ledsavers-mini-campinglampe-p64642>)