

Oniudra - Teknisk rapport

IN1060 - Bruksorientert design

Gruppemedlemmer: Kristian L. Woie, Marit H. Sørli,
Hedda E. Bratt og Mari B. Helland

Vår 2023



Contents

1	Målet med prosjektet	3
2	Presentasjon av video	3
3	Komponenter	4
3.1	Arduino	4
3.2	RGB LED	4
3.3	74hc595	5
3.4	Jump wire F/M	7
4	Krets	7
5	Kode	11

1 Målet med prosjektet

Det overordnede målet med prosjektet har vært å designe en løsning som skal hjelpe eldre med et aldersrelatert problem. Løsningen har blitt designet for, med og av brukere ("DMB"), som er en egen tilnærming innen design hvor de som skal bruke løsningen (brukerne) involveres i designprosessen. Å gjennomføre og lære av en slik DMB-prosess var et viktig underliggende mål i seg selv.

Underveis i designprosessen valgte vi å gå for konseptet "påminnelse" og form-konseptet "slektstre". Den endelige løsningen som presenteres her, er et fysisk produkt i form av et tre som adresserer eldre brukeres behov knyttet til hukommelsestap og opprettholdelse av familierelasjoner.

Løsningen tilbyr en visuell fremstilling av og påminnelse om nære familiemedlemmers bursdager. Treet har bilder av og navn på familiemedlemmer, samt RGB LED-lys som ved bruk av Arduino-teknologi lyser i ulike farger for å vise hvilke familiemedlemmer som har bursdag (i dag), skal ha bursdag i nær framtid (dvs. innen to uker) eller har hatt bursdag i.l.a. den siste uka. Formålet med løsningen er å forbedre Eldres livskvalitet ved å gi en følelse av tilknytning, selvstendighet og minneoppfriskning. I tillegg er tanken at treet kan fungere som et hjelpemiddel og samtalestarter for pårørende og/eller helsearbeidere.

2 Presentasjon av video

Videoen viser løsningen i bruk. Treet har bilder av og navn på 12 familiemedlemmer som er plassert oppover treet i en form som skal representere et slektstre. Brukeren selv er plassert nederst, hennes barn og svigerbarn er plassert over henne og deretter kommer senere generasjoner lenger opp på treet.

Første eksempel viser familiemedlem "Hedda" med en lampe som lyser gult. Dette viser at hun har bursdag i.l.a. de neste to ukene. Deretter går lampen over til å lyse grønt, og dette betyr at hun har bursdag i dag. Etter dette lyser lampen rødt - dette viser at "Hedda" hadde i.l.a. den siste uka. For å få fram poenget med prototypen, har vi manipulert dagene slik at de viser de ønskelige fargene.

3 Komponenter

Dette er de tekniske komponentene vi brukte til prototypen.

Navn	Funksjon	Antall
Arduino Uno Brett	Sender informasjon til kretsen	1
Breadboard	Kobler opp kretsen med shift registerene og LEDene	2
Ledninger	Kobler Arduinoen, LED (<i>F/M</i>) og breadboard (<i>M/M</i>)	48, 33
Motstand	Kontrollerer strømmen fra Arduinoen (270Ω / 220Ω)	36
RGB LED	Indikere bursdager	12
8-Bit Shift Register (74hc595)	Øker antall utganger fra Arduinoen	5

3.1 Arduino

Vi tok utgangspunkt i Arduino Student Kit, men de viktige komponentene skaffet vi selv. Både Arduinoen, de to breadboardene, div ledninger og motstandere ble brukt, samt læringsressursene knyttet til Student Kitet på nettet.

3.2 RGB LED

RGB LED er en spesiell type lysdiode som kan produsere en rekke forskjellige farger ved å blande rødt (Red), grønt (Green) og blått (Blue) lys. LED-en har fire tilkoblingspinner: en felles katode og tre separate anoder for å kontrollere hele LED-enheten og hver fargekomponent individuelt. Ved å justere spenningen eller signalene som sendes til de individuelle fargepinnene, kan vi oppnå ønsket fargemiksing og dermed skape forskjellige farger og lysintensiteter.

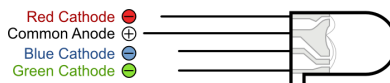


Figure 1: RGB LED, hentet fra https://www.mbeckler.org/microcontrollers/rgb_led/

I vårt prosjekt brukte vi RGB LED-er for å indikere ulike stadier av en persons bursdag. Vi tilordnet fargene gul (verdi: 255, 255, 0), grønn (0, 255, 0) og rød (255, 0, 0) til spesifikke betydninger: Gul indikerte at det var 14 dager eller mindre til noen hadde bursdag, grønn indikerte at personen hadde bursdag den aktuelle dagen, og rød indikerte at personen hadde hatt bursdag for under en uke siden. Ved å bruke disse fargekodene kunne vi visuelt representere bursdagsstatusen til ulike personer.

Disse RGB LED-ene ga oss full kontroll over lysstyrken og fargevalget, noe som var avgjørende for interaksjonen og funksjonaliteten til prototypen vår. Gjennom denne tilpasningsmuligheten kunne vi skape en interaktiv opplevelse for brukerne og tydelig formidle informasjonen om bursdagsstatus.

3.3 74hc595

74HC595 er et 8-bits serietil-parallell skiftregister som har evnen til å konvertere seriell data (bit for bit) til parallell form. Med bare tre kontrollpinner fra Arduinoen, gir 74HC595 oss muligheten til å få 8 ekstra utganger. Det er mulig å koble disse komponentene i parallell, noe som er svært nyttig siden vi ønsker å kontrollere flere enheter samtidig uten å bruke opp alle utgangene på Arduino.

I vårt konkrete tilfelle hadde vi behov for å kontrollere 12 RGB LED-er, hvor hver LED krever 3 forskjellig input for å styre fargekomponentene. Dette betyr at vi i utgangspunktet trenger 36 utganger for å kunne kontrollere LEDene individuelt. Imidlertid, ved hjelp av 74HC595, kan vi oppnå dette med bare 4 utganger fra Arduinoen.

Ved å sende de nødvendige dataene serielt til 74HC595, kan vi konvertere dem til parallell form og styre hver LED ved å velge riktig utgang på komponenten. Dette gir oss muligheten til å oppnå ønsket farge- og lyskontroll for hver eneste RGB LED, selv om Arduinoen ikke egentlig har nok utganger.

74hc595			
Navn	Forkortelser	Beskrivelse	Pin (1 – 16)
Jord	GND	Kobles til jord på Arduinoen	$GND(8)$
Serial Out	Q7	Sender ut dataene fra registeret til neste enhet	$Q7(9)$
Master reset	MR	Nullstiller registeret	$\overline{MR}(10)$
Clock pin	SH_CP	Kontrollerer timingen for å skyve dataene	$SH_CP(11)$
Latch-pinnen	ST_CP	Overfører fra skiftregisteret til lagringsregisteret	$ST_CP(12)$
Output enable	OE	Aktivere eller deaktivere utgangene	$\overline{OE}(13)$
Data input	DS	Dataene servert inn i skiftregisteret	$DS(14)$
Power (Strøm)	VCC	Kobles til strømkilden på Arduinoen	$VCC(16)$
Data output	Q0 - Q7	Data sendes ut til LEDene	$Q0 - Q7(1 - 7, 15)$

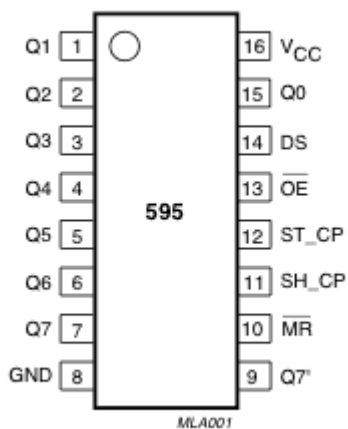


Figure 2: 47hc595, hentet fra Arduinos nettside

Dataen som skal overføres, sendes inn bit for bit i den første komponenten ved hjelp av den serielle datainngangen (*DS*). Når hver bit av dataen er sendt til denne inngangen, pulserer skiftregisterklokken. Ved å pulsere denne klokken, blir hver bit forskjøvet fra en registercelle til neste i serien av 74HC595-enheter. Deretter pulserer lagringsregisterklokken (*ST_CP*) og forårsaker at de skiftede dataene blir "låst" i lagringsregisteret. Da er dataen tilgjengelig på hver av utgangene. Selve utgangene til hver 74HC595-enhet er aktivert ved å kontrollere utgangsaktivering (*OE*). Når *OE* er lav, er utgangene aktivert. Serielle datautgangen (*Q7*) på den første 74HC595-enheten er koblet til serielle datainngang (*DS*) på den neste 74HC595-enheten i kjeden. Dette gjør at vi kan sende dataen 001010010100... hvor det vil gå strøm gjennom pin nr. 2, 4, 7, 9, osv.... Vi bruker dette for å få akkurat de LEDene vi ønsker til å lyse i ønskelig farge og samtidig ha så mange vi vil.

74HC595 fungerer som en effektiv mellomkobling mellom Arduinoen og de mange enhetene vi ønsker å kontrollere. Den gir oss en praktisk og plassbesparende løsning som lar oss utvide kapasiteten til Arduinoen og samtidig oppnå ønsket funksjonalitet.

3.4 Jump wire F/M

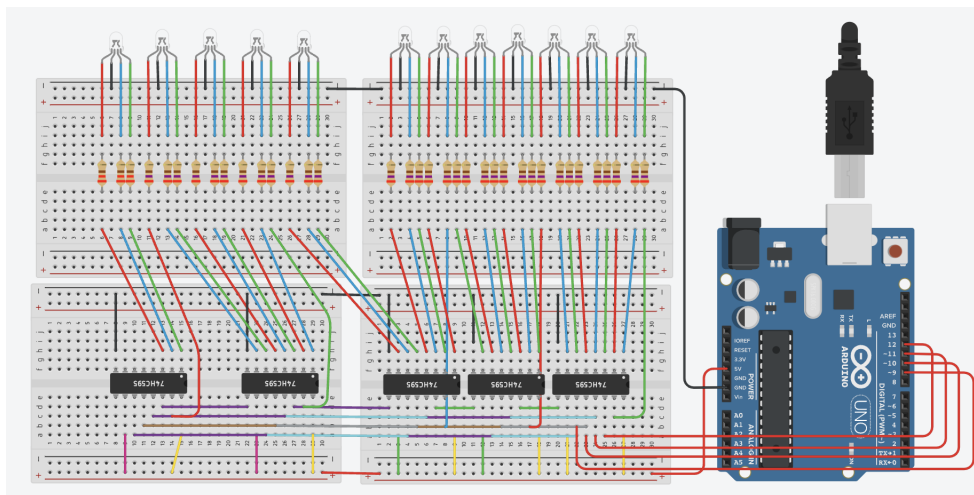
For å oppnå en estetisk tiltalende og praktisk løsning valgte vi å feste LED-ene direkte i planten ved hjelp av jumper wires F/M i stedet for å plassere dem på breadboardet. Dette gjorde det mulig å sømløst integrere LED-ene i plantens naturlige omgivelser. Ved å tilkoble den ene enden av jumper wires til LED-pinnene og den andre enden til breadboardet, oppnådde vi en sikker og stabil tilkobling uten synlige kabler. Denne tilnærmingen bidro til en mer elegant og funksjonell montering av LED-ene i planten.

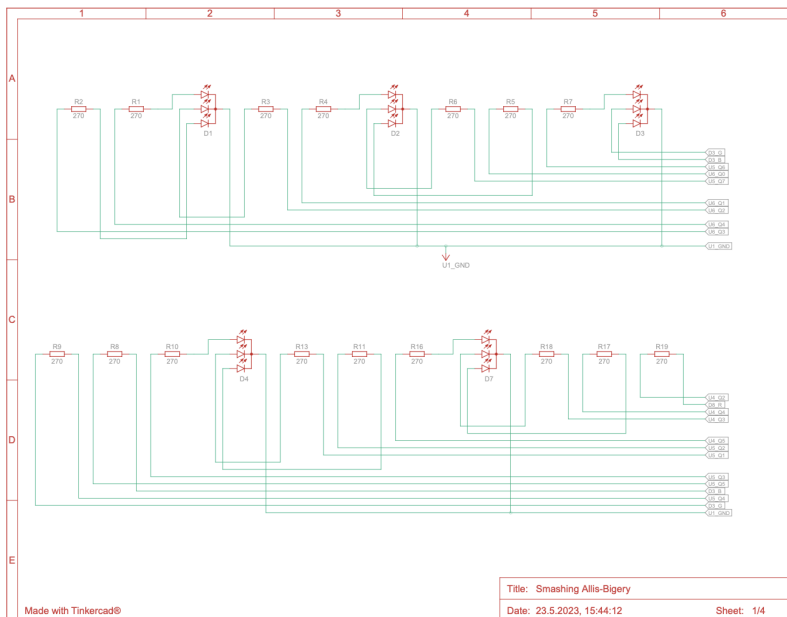
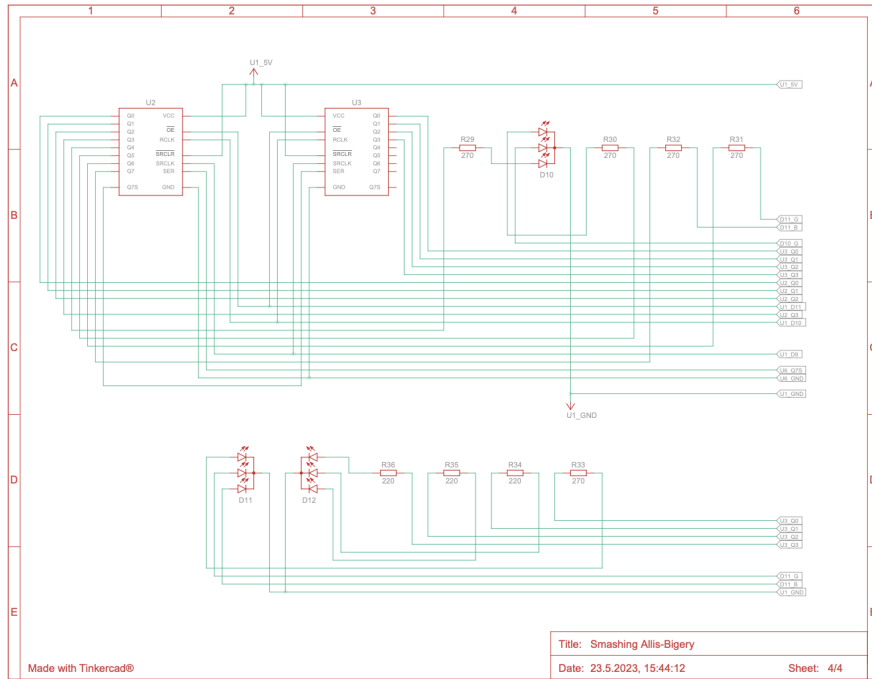
4 Krets

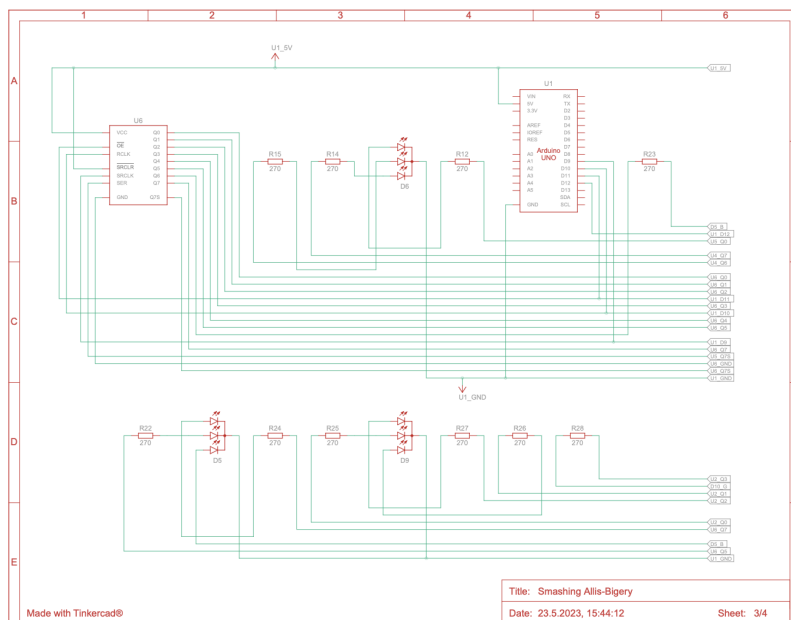
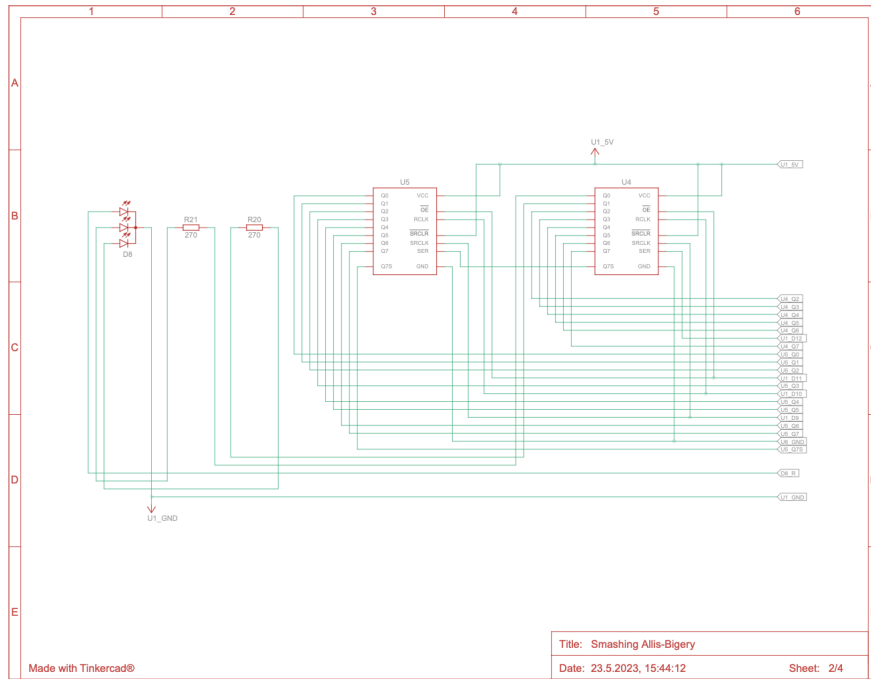
Før vi kjøpte inn komponenter til den fysiske kretsen, lagde vi den digitalt i Tinkercad. Vi hadde noen tekniske problemer ang kommunikasjonen mellom datamaskinene og Arduinoen, men med denne løsningen klarte vi å både lage koden og kretsen før vi trengte å sette det sammen i virkelighetet. Dette lot oss også dobbeltsjekke at komponentene ikke blir ødelagt eller skadet, og om alt fungerte som det skulle.

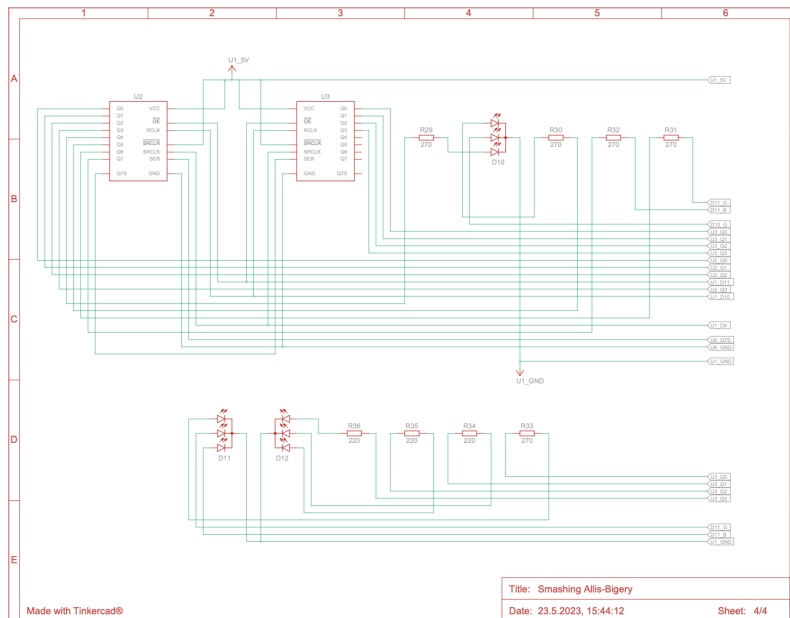
Kretsen består av tre faser. Først sendes informasjon fra Arduinoen til riktig skift register og videre til riktig pin. Deretter sendes den ut gjennom motstandene og til RGB pinnen som gjør at LEDen lyser i riktig farge. Til slutt går signalene tilbake til jord på Arduinoen som gjør at vi har en lukket krets.

Vi har brukt både 220Ω og 270Ω , men dette er for liten forskjell på dem til at man merker forskjell. Noen av lysene skulle også festes lengre opp enn andre, så vi brukte to jump wires per pin for å forlenge ledningene. Ellers er kretsen i Tinkercad identisk med den faktiske kretsen vår. Under ser man først oppsettet og deretter de skjematiske kretsene fra Tinkercad.









5 Kode

Vi har hatt noen problemer med koden som vi ikke klarte å løse. Noen av LEDene vil ikke skru seg av selv om vi implisitt setter verdiene til pinnene til LOW, eller på om vi setter verdien til HIGH. Fagrebeholderene (redPin, greenPin og bluePin) virker som om de har byttet om verdi uten at vi finner ut hvordan. Hverken å endre shiftregister pinnene, navnene eller verdiene fungerer og vi har prøvd å fjerne fysiske ledninger for å sjekke om LEDen ikke er ødelagt. Ingen ting av det vi har prøvd ser ut til å gjøre noe forskjell. Den LEDen som indikerer at noen har bursdag om 14 dager, blir lilla om vi setter grønn og rød til HIGH, men gul om vi setter rød og blå til HIGH. Dette gir ikke mening siden den gule RGB-koden er (255, 255, 0). igjen har vi prøv alle mulig kombinasjoner av feilsøking og kommer heller ikke fram til en løsning.

25.04. er satt inn som en eksempeldato fordi det var et av tidspunktene hvor to av familiemedlemmene hadde bursdag rimelig nærme. Vi hat ikke tatt hensyn til månedsskifte eller at det kommer noen nye til i familien.

```

// Oniudra - IN1060

// Datoen her er satt til 24.04. slik at et LED skal lyse gult
// og et skal lyse rødt

// ps: for å få resultat fortere, prøv å endre "doegn"
// til noe mindre (linje 29)

// setter utgangene fra arduinoen til shift registorene
int dataPin = 12;
int enablePin = 11;
int latchPin = 10;
int clockPin = 9;

// lager hjelpeverdier til indexsering av pinner
int number_of_74hc595s = 5;
int numOfRegisterPins = 36;
int numRGBLeds = 12;

// henter antall pinner
bool registers[36];

// lagrer pinene med samme farge i lister
int redPin[] = {0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33};
int bluePin[] = {1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34};
int greenPin[] = {2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35};

// venter et døgn mellom hver oppdatering
long dogn = 24 * 60 * 60 * 1000;

// holder styr på dag, måned og pinindex til hver av personene
struct Birthday {
    int day;
    int month;
    int ledIndex;
};

```

```

// bursdagene til personene involvert
Birthday birthdays[] = {
// dag | mnd | pin
  {31, 12, 1},
  {21, 6, 2},
  {22, 1, 3},
  {27, 1, 4},
  {30, 7, 5},
  {25, 4, 6}, // har bursdag dagen etter (gult)
  {20, 4, 7}, // hadde bursdag for 4 dager siden (rødt)
  {20, 7, 8},
  {12, 12, 9},
  {18, 2, 10},
  {10, 10, 11},
  { 2, 5, 12},
};

void setup() {
  // aktiverer shift registerene slik at de kan ta imot data
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(latchPin, OUTPUT);
  pinMode(enablePin, OUTPUT);

  // aktiverer pinne
  digitalWrite(enablePin, LOW);

  // initialiserer alle lysene til å være av
  for (int i = 0; i < numOfRegisterPins; i++) {
    registers[i] = LOW;
  }
}

// hovedprogrammet
void loop() {

```

```

// lager en dato
int day = 24;
int month = 4;

// starter med å slå av alle lys for å ikke
// overbelaste shift registerene
turnOffLEDs();

// løper gjennom alle lysene og oppdaterer fargene dersom
// det er nødvendig
for (int i = 0; i < sizeof(birthdays) / sizeof(birthdays[0]); i++) {

// lager en variabel for på finne antall dager mellom dagens
// dato og bursdagen
int days_to_bday = birthdays[i].day - day;

// endrer lyset til gult dersom bursdagen er om under 2 uker
if ((days_to_bday <= 14) &&
    (birthdays[i].month == month)) {
// Slår på rød farge
turnLedYellow(birthdays[i].ledIndex);
}

// endrer lyset til rødt dersom burdagen var for 7 dager siden
if ((days_to_bday >= -7
    && days_to_bday < 0) && (birthdays[i].month == month)) {
// Slår på gul farge
turnLedRed(birthdays[i].ledIndex);
}

// endrer lyset til grønt dersom det er dagen personen har bursdag
if ((birthdays[i].day == day) && (birthdays[i].month == month)) {
// Slår på grønn farge
turnLedGreen(birthdays[i].ledIndex);
}
}
}

```

```

    // oppdaterer dette hvert døgn for å ikke overbelaste CPUen
    delay(dogn);
}

// funksjoner brukt til å endre fargene på LEDene

// skriver data til shift registorene og setter dem til riktig pin
void registerWrite(int pin, int value) {
    digitalWrite(enablePin, LOW); // deaktiverer enablePin
    digitalWrite(latchPin, LOW); // forbereder dataoverføring
    digitalWrite(clockPin, LOW); // klar til å motta data

    int val = registers[pin]; // Leser nåværende verdi fra
                               // registers-arrayet basert på pin
    digitalWrite(dataPin, val); // Sender verdien til dataPin (DS)
                                // for å skrive til registeret
    digitalWrite(clockPin, HIGH); // overfører data til registeret
    digitalWrite(latchPin, HIGH); // låser dataene i registeret

    registers[pin] = value; // Oppdaterer verdien i registers-arrayet
                            // med den nye verdien
}

// Skrur av alle ledene
void turnOffLEDs() {
    for (int i = 0; i < numRGBLeds; i++) {
        registerWrite(redPin[i], LOW);
        registerWrite(greenPin[i], LOW);
        registerWrite(bluePin[i], LOW);
    }
}

// Slå på riktig rødt lys
void turnLedRed(int led) {
    registerWrite(redPin[led], HIGH);
}

```

```
    registerWrite(bluePin[led], LOW);
    registerWrite(greenPin[led], LOW);
}

// Slå på riktig grønt lys
void turnLedGreen(int led) {
    registerWrite(redPin[led], LOW);
    registerWrite(bluePin[led], LOW);
    registerWrite(greenPin[led], HIGH);
}

// Slå på riktig gult lys
void turnLedYellow(int led) {
    registerWrite(redPin[led], HIGH);
    registerWrite(greenPin[led], LOW);
    registerWrite(bluePin[led], HIGH);
}
```