



INSTITUTT FOR ELEKTRONISKE SYSTEMER

IELS1001 - INGENIØRPROSJEKT

LED-planke

August 4, 2023

Hva er LED-planken?

LED-planken er en addresserbar RGB LED-stripe som ligger i solid beskyttelsesplast. På toppsiden av planken er det nummerert leddene fra 0 til 31. I tillegg har hver LED lagt ved sin binærverdi og heksagonal representasjon. Formålet med dette er at man skal kunne bruke LED-planken som en visuell representasjon av digitale bits.

På planken er det to tilkoblingspunkter, en som er inngangen til stripen og en som er utgangen. Disse tilkoblingspunktene er 6x2 konnektorer, men vi benytter kun de fire i midten på nederste rad.

På inngangssiden, siden ved 0 tallet, er det fire tilkoblinger. Disse er fra venstre: Jord, Digital IN, Spennning (5V) og Digital OUT. (Se figur 1)

På utgangssiden, altså siden ved 31 tallet, er det tre tilkoblinger. Disse er fra høyre: Jord, Digital OUT og spenning (5V). (Se figur 1)

Formålet med å ha tilkoblinger på begge sidene er at man kan koble en ny LED-planke på enden for å utvide planken. Dette gjøres ved å koble Digital OUT fra en planke til Digital IN på en annen.

Som dere kanskje ser finnes Digital OUT på begge sidene, dette er slik at man kan istedet koble plankene på rad. Konseptet for å utvide LED-plankene er de samme. Koble planke 1 sin Digital OUT til planke 2 sin Digital IN.

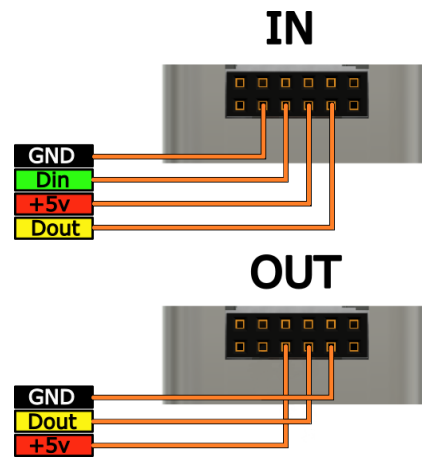


Figure 1: Pinout for LEDplankene

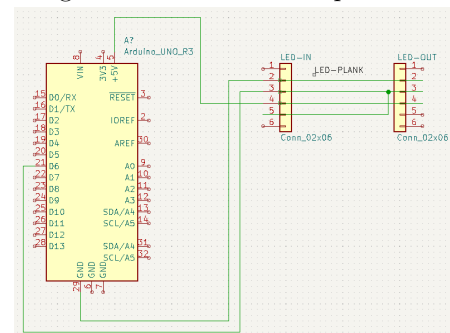


Figure 2: Oppkoblingsskjema, husk at konnektorene nederste rad

Hvordan bruke LED-planken?

LED-planken kan kobles opp mot en mikrokontroller, som en Arduino UNO eller en ESP32. På mikrokontrolleren tar vi i bruk biblioteker for å lettere styre LED-stripen. Siden LEDene er adresserbare kan bibliotekene bestemme egne parametre til de forskjellige LEDene basert på indeksen til LEDen.

Adafruit Neopixel

Det biblioteket vi skal bruke heter [Neopixel](#) og er laget av Adafruit Industries. Neopixel biblioteket legger til flere enkle funksjoner som tillater deg å kontrollere en Neopixel-kompatibel LED-stripe, som LED-planken er. Det anbefales å ta en titt på referansen til biblioteket, lenket over, og se på et par av eksemplene for å se hvordan biblioteket kan brukes. I de fleste eksempelkodene kobles planken til pinne 6 på UNOen.

Oppgave: Småoppgaver

Oppgave: Lag mønster

Ved bruk av Neopixel biblioteket er det enkelt å kontrollere fargen til de forskjellige LEDene og lage kule mønster. I eksempelet "strandtest" kan dere se flere forskjellige varianter av dette, og hvordan det er gjort.

Prøv selv å lag et unikt mønster. Dette kan enten gjøres ved kun ved intern programlogikk (som løkker), eller så kan man legge til eksterne kretskomponenter for å styre mønsteret (som med et potensiometer).

Oppgave: Seriekobling av planker

Som beskrevet i dokumentasjonen kan LED-plankene serie-kobles for å utvides. Dette begrenses kun av programkoden og elektrisiteten selv. Det er 32 LEDs per planke. Gå sammen med andre og prøv å koble mange planker på rad.

Hva tror dere kommer til å skje med LED-planken? Hvor langt tror dere planken kan gå før det begynner å påvirke planken? Hvorfor oppfører planken seg slik?



Som sagt vil programkoden begrense hvor langt LED-lenken kan gå. For å koble flere LEDs må man definere hvor mange LEDs programmet skal skrive til. I tillegg er de fleste eksempelprogrammer endret til at lysstyrken er på 1/5 av maks (50 av 255). I hvor stor grad tror dere dette kommer til å påvirke lengden.

Program: Binær kalkulator

Lenke til kode: [Github](#)

Koden over tilhører et program hvor man får representert addisjon og subtraksjon med binære tall. Som dere kanskje vet er binære tall fenomenale til å representere store mengder tall, og kan representere opp til flere milliarder tall med bare 32 siffer. I tillegg er operasjonene veldig enkle å forstå for maskiner.

Oppkoblingen er avbildet på figur 3. Planke 1 kobles Digital IN til pinne 6. På planke 2 og 3 kobles Digital IN til Digital OUT på den forrige planken, altså 2 til 1, og 3 til 2. Konseptet er at man kobler Digital OUT fra forrige planke inn i Digital IN på neste. For å få en mer lesbar og korrekt representasjon av tallene er det anbefalt å snu plankene opp ned og sette den første planken øverst (som avbildet på figur 3). Dette er fordi binærtall ofte representeres med Least Significant Bit til høyre, noe som strider med indekseringen på planken.

Programmet fungerer ved at man skriver inn en utregning i Serial-konsollen, for eksempel "101 + 202". Planke 1 vil da lyse opp med binærtallet for det første tallet, planke 2 vil lyse opp for det andre og planke 3 vil lyse opp med svaret av utregningen. Figur 4 viser hvordan dette skal se ut ved regnestykket "101 + 202". For å kontrollsjekke tallet kan du summere alle LEDs som lyser hvit med den korresponderende binærverdien som er skrevet på LED-planken.

Oppgaver:

- Prøv å finne det største mulige tallet planken kan representere.
- Prøv å se hva som skjer om man går over det tallet
- Prøv med negative tall og se hvordan programmet håndterer dette

Poenget med disse oppgavene er å gi en visuell representasjon av binær logikk, men også å vise et potensiell bruksområde for planken.

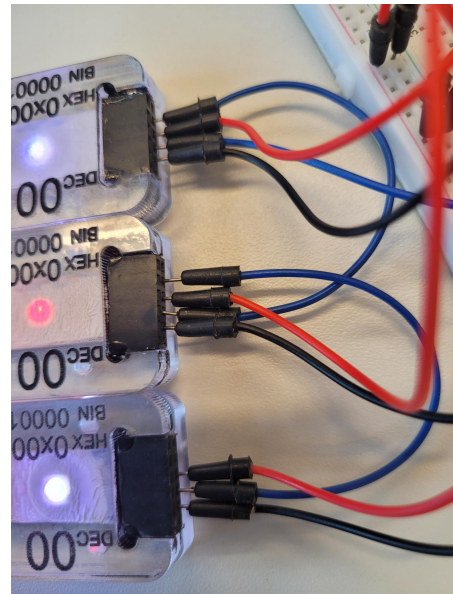


Figure 3: Øverste planke er den som er koblet til UNOen



Figure 4: $101 + 202 = 303$

Program: Gray-code

Lenke til kode: [Github](#)

Koden over tilhører et program som visuelt representerer hvordan GRAY-code fungerer. Gray-code er en form for binærtallkode som skal i flere tilfeller være mer effektiv for maskinger å utføre. Dette er fordi de kun endrer et siffer hver gang det går opp et tall, sammenlignet med vanlige binærtall hvor det kan endres flere samtidig.

Oppkoblingen, se figur 5, er en knapp koblet til pinne 4 på UNOen, og LED-planken koblet opp til pinne 6, som tidligere. Knappen er av typen pull-down. Ved å trykke på knappen, eller holde den inne, øker tallet fram til den satte begrensningen. I likhet med binærkalkulatoren må planken snus for å få best mulig representasjon.

I koden er det mulig å kommentere ut GRAY-Code konversjonen, slik at man istedet ser binærtall.

Det er også mulig å øke og senke grensen, som er nå satt til 255 (2^8-1)

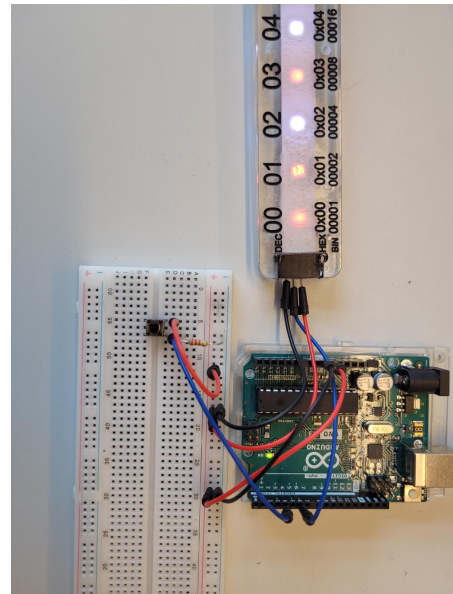


Figure 5: Bilde av oppkoblingen

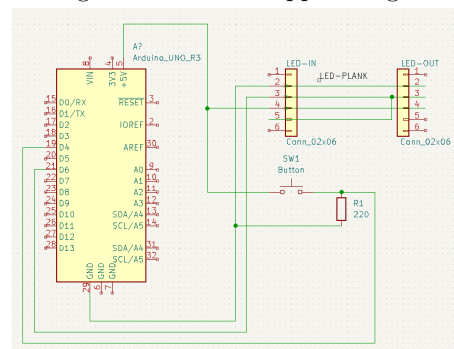


Figure 6: Oppkoblings-skjema

Oppgave: Spill

Dere skal nå bruke LED-planken til å lage et form for spill. Kravene for spillet er:

- Spillet skal bruke minst en LED-planke
- Spillet skal bruke minst 2 styreknapper
- Spillet skal ha minst to vanskelighetsnivåer
- Spillet skal ha unik animasjon for tap og vinn
- Spillet skal ta vare på og formidle poengsum til spilleren
- Koden skal være ikke-blokkerende

TIPS:

- Dere står fritt fram til å slå sammen flere LED-striper og samarbeide, så lenge alle deltar og forstår hvordan koden fungerer.
- Vanskelighetsnivåer kan være enten valgbare eller at spiller blir vanskeligere når man spiller.