NTNU

elsys
ELEKTRONISK SYSTEMDESIGN & INNOVASJON

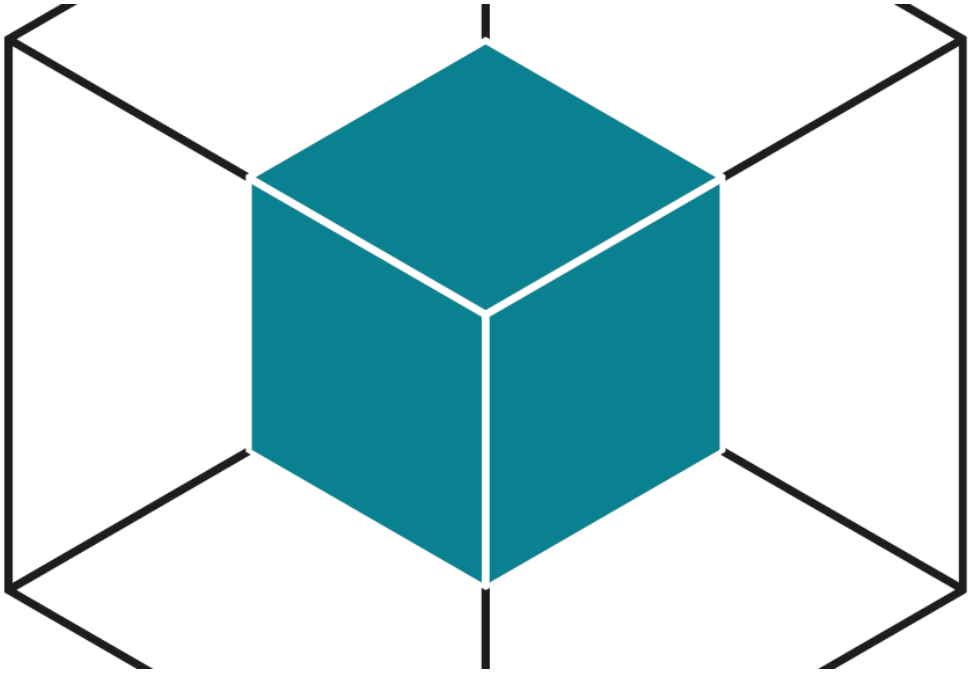
# TTT4255 Elektronisk systemdesign, grunnkurs

# A2: DC-motor

Elektronisk systemdesign og innovasjon

Ida Bjørnevik, Sven Amberg, Amalie Fridfeldt Hauge og Peter Magerøy

28.06.2023



## Innhold

**Introduksjon** **2**
Teori . . . 2

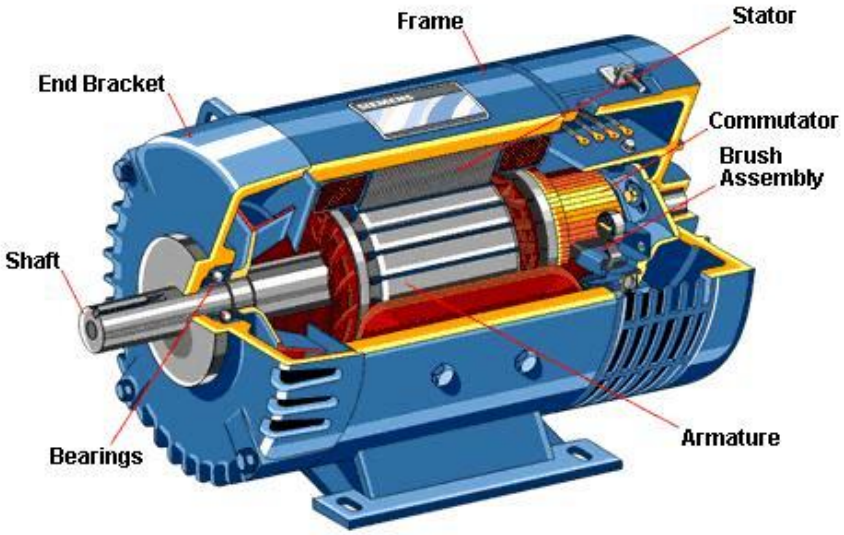
**Prosjektet** **3**
Utstyrsliste . . . 3
Steg 1: Oppkopling av krets. . . . 3
Steg 2: Koble til GPIO . . . 3
Steg 3: Programmering . . . 5
Steg 4: Intervall . . . 6

**Konklusjon** **7**

## Introduksjon

Passer for deg med **ingen forkunnskaper.**

I denne modulen skal dere bli kjent med DC-motorer, se figur 5, og lære hvordan man kan styre de ved hjelp av ESP32-S2.




**Figur 1:** Illustrasjon av en likespenningsmotor.

**Teori for den interesserte**

DC-motorer er elektriske motorer som roterer med hastighet gitt av spenningen. DC-motorer bruker magnetiske felt som oppstår rundt elektrisk strøm. Det magnetiske feltet får rotoren til å rotere. Rotasjonen og hastigheten er avhengig av både spenning og motordesignet. Motoren dere har fått utdelt har makshastighet ved 6 V på 100 RPM (rotasjoner per minutt).

Her er noen lenker dersom du er interessert i å lære mer om DC-motorer (ikke nødvendig for å gjennomføre oppgavene).

- **How does an Electric Motor work? (DC Motor)** `https://youtu.be/CWulQ1ZSE3c`
- **What is PWM?** `https://www.youtube.com/watch?v=B_Ysdv1xRbA&t=79s&ab_channel=HackTheWorld`

# Prosjektet

## Utstyrsliste

- DC-motor
- ESP32-S2
- Papirark (frivillig)

### Steg 1: Oppkopling av krets.

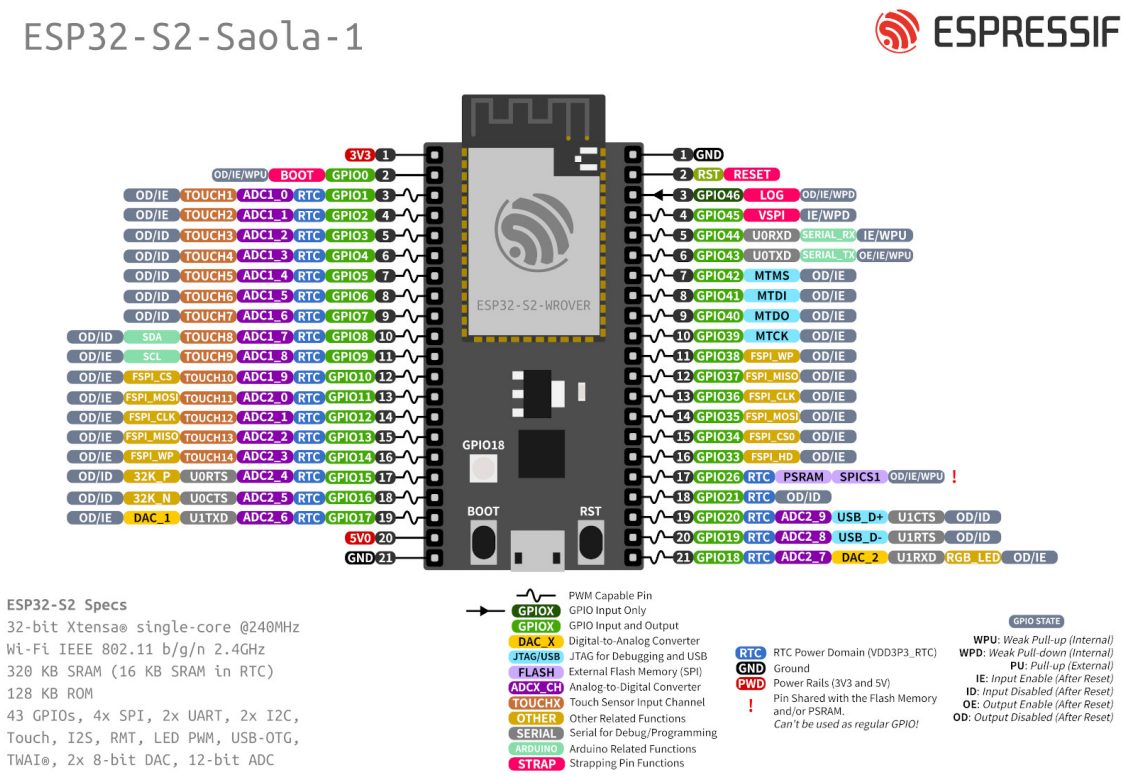
Koble kretsen etter kretsskjem i i figur 2, det er lagt ved et bilde av den fysiske kretsen i figur 5 dersom du står fast. På DC-motoren er den røde ledningen positiv og den svarte er negativ.


**Figur 2:** Kretsskjema for DC-motor kopla til ESP32.

Uten å laste opp kode, koble ESP-en til PC med USB. Hva skjer? Prøv å bytte til 3,3V ved å koble den røde ledningen til 3,3 V i stedet for 5 V. Hva skjer? Lag gjerne en arm i papir for å lettere se hvordan den roterer.

### Steg 2: Koble til GPIO

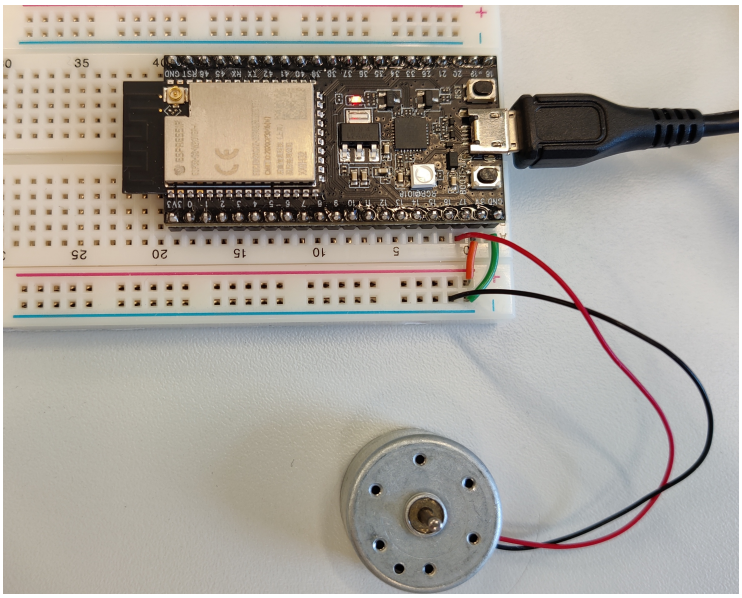
For å få motoren til å rotere med ulik hastighet bruker vi PWM (pulsbreddemodulasjon). Det er derimot ikke alle pinner på ESP-en som har mulighet til å lage et PWM-signal. ESP-en dere har fått utdelt er et utviklingskort kalt ESP32-S2-Saola-1 og har pinout vist under. Her kan dere legge merke til pinnene som har en bølge mellom pinnetall og kort i stedet for en rett strek. Disse pinnene kan vi bruke for å lage et PWM-signal og det må derfor velges en av disse for å kunne styre hastigheten til motoren.




**Figur 3:** Oversikt over pins for ESP32-S2.

Koble den røde ledningen til GPIO17 som vist i figur 4. Husk å merke dere at GPIO17 ikke er det samme som pinne nummer 17 på utviklingskortet. Dette gir oss muligheten til å styre hastigheten til motoren med PWM.


**Figur 4:** Kretsskjema for DC-motor koblet til ESP32.



**Figur 5:** DC-motor tilkoblet ESP32-S2.

### Steg 3: Programmering

For å kontrollere hastigheten bruker vi som nevnt PWN. Skriv av kodesnuttene underveis.

Før setup-funksjonen definerer vi variablene i programmet. Se figur 6.t.

```
const int DCmotor = 17; // 17 vil si GPIO 17
// Setting PWM properties
const int freq = 250; // Frequency
const int pwmChannel = 0; // Which of the PWM-channels we use
const int resolution = 8; // Resolution of the signal
```

**Figur 6:** Variabeldefinisjoner i programmet.

I *setup()* setter vi opp et PWM-signal, se figur 7. Her trenger dere ikke forstå hva som skjer for å kunne gjenbruke dette senere.

```
void setup(){

  // Configure PWM functionalities
  ledcSetup(pwmChannel, freq, resolution);

  // Attach the channel to the GPIO to be controlled
  ledcAttachPin(DCmotor, pwmChannel);
}
```

**Figur 7:** Setup-funksjonen i programmet.

Loop-funksjonen kjører igjen og igjen så lenge ESP32 er på. I funksjonen, se figur 8, justerer vi opp og ned rotasjonshastigheten.

```
void loop() {
  // Increase the speed
  for(int dutyCycle = 0; dutyCycle <= 255; dutyCycle++) {
      // Changing the rotation speed with PWM
      ledcWrite(pwmChannel, dutyCycle);
      delay(15);
    }
    delay(2000);

    // Decrease the speed
    for(int dutyCycle = 255; dutyCycle >= 0; dutyCycle--) {
      // Changing the speed PWM
      ledcWrite(pwmChannel, dutyCycle);
      delay(15);

    }

}
```

**Figur 8:** Loop-funksjonen i programmet.

### Steg 4: Intervall

**Tenke selv**

Endre på ulike parametere og få motoren til å rotere på raskeste innstilling i 10 s med 5 s pause mellom.

## Konklusjon

I denne modulen har du lært å koble opp en DC-motor til en mikrokontroller og styre rotasjonshastigheten ved hjelp av pulsbreddemodulasjon. Denne kunnskapen kan man bruke for å få hjulene på en modellbil til å rotere, lage vifte og mye mer. Kunnskap om PWM kan også brukes i alt fra å kontrollere servoer til å lage direksjonelle høyttalere.

**Refleksjonsspørsmål**

- Hvilke andre ting kan du bruke en DC-motor til?
- Hva er fordelen med å gi navn til pinner i koden?
- Prøv å forklar hvordan kretsen fungerer med egne ord.