

## Prosjektoppgaver IELET2111 vår 2023

Hver gruppe kan velge **en** av de fire oppgavene presentert under, eller en forhåndsgodkjent egen oppgave.

For oppgaver med lav utfordringsgrad vil det trolig være vanskeligere å vise tilstrekkelig faglig kunnskap til å oppnå høyeste karakter, men er til gjengjeld enklere. Men ved å legge til ekstra funksjonalitet. Merk at utfordringsgraden er kun et estimat.

Oppgaver med høy vanskelighetsgrad vil kreve mye innsats og har en viss fare for å ikke nå helt i mål. Derfor kreves det god prosjektstyring og modularitet i løsningen av oppgaven for å komme være sikker på at arbeidet som leveres er av høy kvalitet.

Skulle omfanget i oppgaven vise seg å være for stor vil karakterskalaen justeres for å ta hensyn til det.

Bruk teknikker for prosjektstyring og organiser arbeidsoppgavene godt i gruppen og sikre at koden virker. Det vil bli vektlagt hvor lesbar, selvforklarende og godt organisert koden er. Og i hvilken grad kode er gjenbrukbar.

Leveransen av prosjektet skal være i form av prosjektrapport som leveres mot slutten av semesteret (dato og utformingsveiledning kommer). I tillegg vil det bli gjennomført en muntlig justerende eksamen hvor alle i gruppen kan bli spurt om alle aspekter med koden og løsningen.

### Oppgave 1 - Servokontroller

Man skal utarbeide koden for et system som kan styre en servo i et spjeld, samt legge til rette for feilprediksjon.

#### Estimert utfordringsgrad: 1

- MÅ: Styring av servo men justerbar hastighet og posisjoner
- MÅ: UART har menysystem for innstillinger.
- MÅ: Sensor på strømtrekk for å kompensere for treghet i spjeld og prediksjon av slitasje.
- BØR: Modulinngang for i2c eller bryter-styring.
- KAN: Ekstra mulighet for LCD og knappe-meny.

### Oppgave 2 - Viftekontroller

Et system for styring av kraftige vifter som brukes for avkjøling av et datasystem. Det skal i utgangspunktet ha en grad av feilhåndtering.

#### Estimert utfordringsgrad: 2

- MÅ: PWM-styring av inntil 8 vifter.
- MÅ: Monitorering av turteller for prediktering av tidlig feil.
- MÅ: Lagring av trend-data for prediksjon av feil.
- MÅ: UART har menysystem for innstillinger.
- BØR: I2C master for temperaturmåling
- KAN: I2C slave for rapportering videre.
- KAN: Slave-adresse settes via menysystemet.
- KAN: Mulig utvidelse til full styring av parameter via I2C.

### Oppgave 3 - I2C slave for monitorering av spenning, temperatur og viftehastighet via lys-detektor.

Et system knyttet til viftekontrolleren i oppgave 2. Det skal monitorere spenning, temperatur og viftehastighet (ved hjelp av lys-detektor).

#### Estimert utfordringsgrad: 2

- MÅ: Lage en I2C slave, enten egenkodet eller via rammeverk.
- MÅ: Noden skal overvåke både egen og ekstern spenning
- MÅ: Noden skal overvåke temperatur via en termistor.
- MÅ: Noden skal med hjelp av lys- og fotodiode bestemme hastigheten til ekstern vifte.
- MÅ: Via feilutganger vil noden varsle både høy temperatur, og gi feil dersom begge viftene står stille i for lang tid, eller temperaturen er for høy.
- MÅ: Endringer i innstillinger må kunne gjøres over I2C og lagres på kontrolleren.
- BØR: Viften skal overvåkes over tid, slik at avvik i viftehastighet og evt varsler kan leses av via I2C.
- Tilleggskreter kan brukes etter behov. EEPROM / ADC / OPAMP osv.

### Oppgave 4 - Demonstrator av blitter for AVR

En blitter er en dedikert krets for rask forflytting/endring av data i en datamaskins minne. I dette prosjektet skal man demonstrere dette ved å bruke en ATmega169A på et spesialdesignet kretskort sammen med ekstern SRAM og 5 stk 74-logikk tellere for å akselerere grafikk på en 320x240 RGB LCD-skjerm.

#### Estimert utfordringsgrad: 3

- MÅ: UART for kommunikasjon til omverden. Grafikk-kommandoer og data kommer over UART.
- MÅ: Definere kommandoer over UART for å tegne grafikk og tekst.
- BØR: Lagring av grafikk i FLASH.
- BØR: System-kontroll ved start.
- KAN: Lage et scriptingspråk med makroer som kan lagres i ekstern FLASH for enerering av grafikk.
- KAN: Presse ytelsen i systemet.
- KAN: Krever kanskje utvidet bruk av pekere som funksjonspekere
- KAN: Krever kanskje innslag av assembly i høynivåkode. Selvstudium.