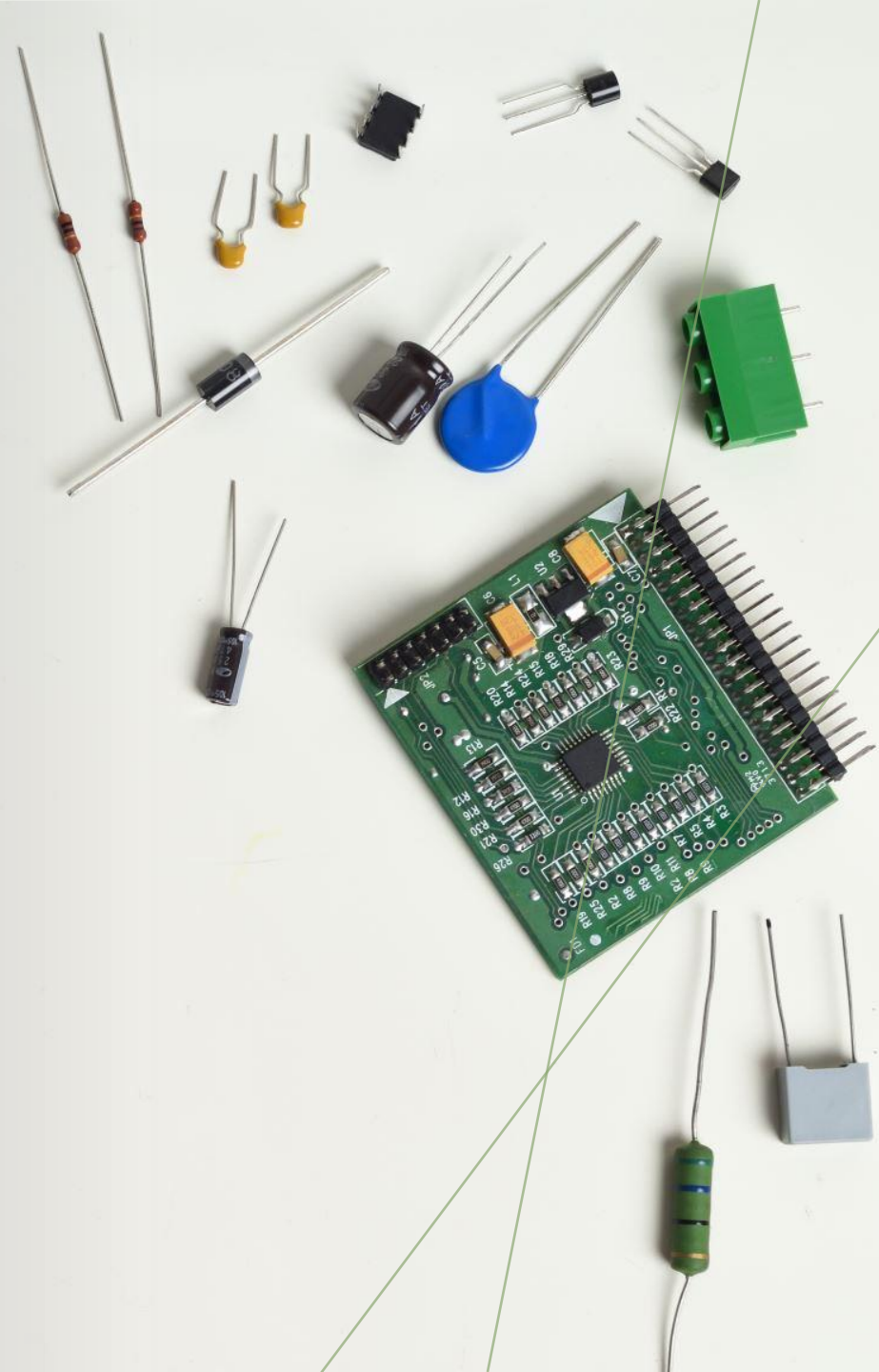


PROSJEKT IELET2111



INNHold

Generelt oppsett

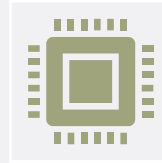
ADC og interrupt

Timere og PWM

Seriell kommunikasjon

Ideer til bruk og videre utvikling

GENERELT OPPSETT: KJØLEPROGRAM



To sensorer: Potensiometer og temperatursensor, en motorvifte.



ADC for måling av sensorverdier.



Temperatur printes til serial monitor via USART.



Dersom potensiometeret går over 30% av maksverdi skrus viftemotoren seg på.



Timer som styrer PWM for å drive viftemotoren.



Verdien på temperatursensoren styrer hastigheten på viften.

ADC OG INTERRUPT

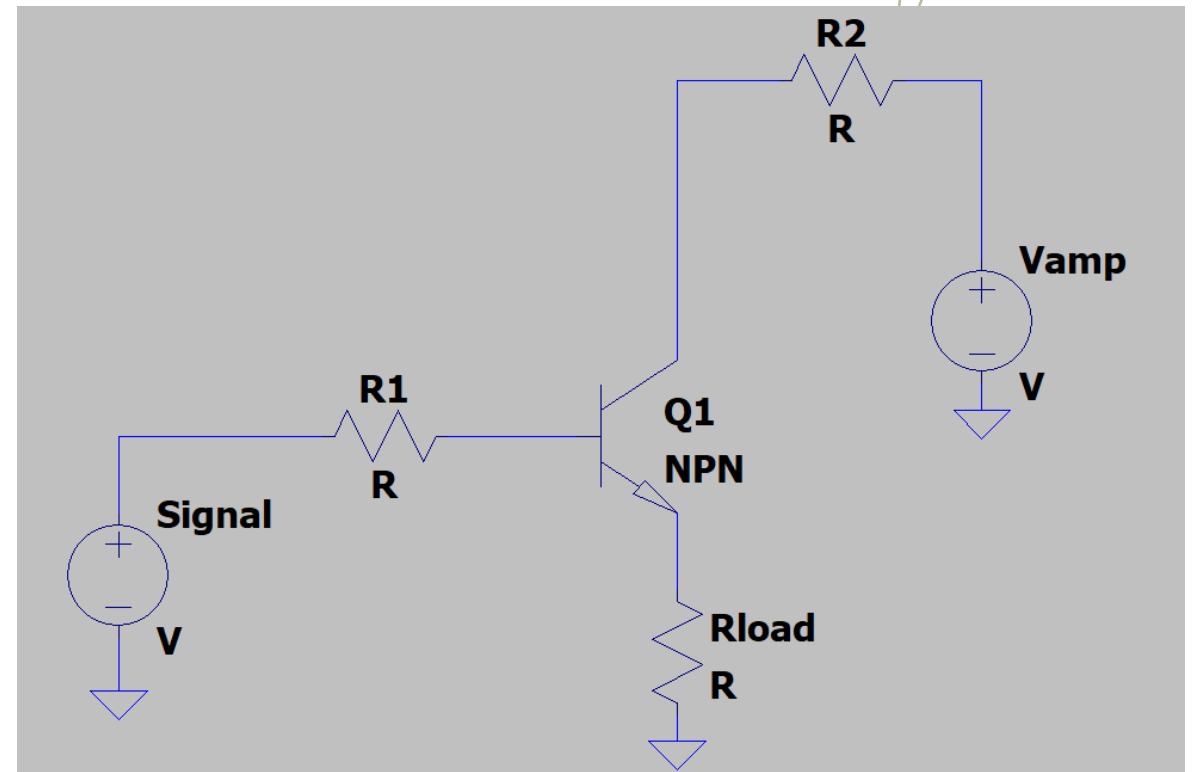
- Analog to digital converter nødvendig for å lese sensorene.
- ADC kan kun lese en sensor av gangen. Toggler mellom de to sensorpinnene etter hver avlesning.
- Interruptfunksjon trigges med ADC_vect. Dette betyr den trigges når ADC-konverteringen er ferdig, altså ved hver konvertering.
- Switch-case inne i interrupt-funksjonen. Sjekker hvilken pin som er blitt lest av og lagrer avlesningsverdi. Toggler inngangspinne før neste avlesning.
- Leser av sensorverdi med ADC-registeret. Bruker right adjusting resultatbits som default.
- Bruker prescaler på 128, dette gir 125 kHz.

HVORFOR DENNE LØSNINGEN?

- Finner ingen løsning der ADC kan kjøre parallelt med to sensorer. Derfor må hver sensor leses etter hverandre.
- Interrupt er hardware-basert. Kan derfor ikke bruke spesifiserte tilfeller for å trigge interrupten som en if-else-løsning. Kan kun bruke interrupt-vektorene som eksisterer fra før av.
- ISR(ADC_vect) trigges når konverteringen er ferdig. Må derfor sortere hvilke verdier som er relevante *etter* interrupten.
- Mest praktisk å da lese hvilken pinne som trigget interrupten og sortere basert på dette.
- Anbefalt å ha så korte ISR-funksjoner som mulig. Switch-case-strukturen er likevel relativt kort.
- Anbefalt å bruke frekvens på 50-200 kHz for full oppløsning. Lavere frekvens betyr mer nøyaktighet.

TIMERE OG PWM

- Timere og PWM brukes for å drive viftemotoren
- En timer er nødvendig for å generere et PWM signal på OCR0A
- En timer lar oss variere PWM signal fra 0V-5V
- De fleste vifter drives på 12V, ikke 5V (--)



HVORFOR DENNE LØSNINGEN?

- Man vil kunne variere utgangsspenning basert på temperatur
- Lite elektronikk/kode er nødvendig
- Kan drive så mange vifter som nødvendig (realistisk sett) i et system
- Samme utgangssignal kan brukes ved flere spenningsnivåer på grunn av forsterkerkrets
- Lett å bruke, håndtere og modifisere

SERIELL KOMMUNIKASJON

- USART nødvendig for å kommunisere mellom ATmega328P og PC
- Det er lagd et bibliotek for å sende og motta tekst og tall mellom -32,768 to 32,767
- Funksjonene for tekst bruker pekere for å iterere gjennom minneadressene
- Alle funksjonene er avhengig av NULL-terminering på en eller annen måte
- Valgt en BAUD-rate på 9600
- Bruker 8 data bits og 1 stop bit
- ATmega sender temperatur som kan avleses i seriellmonitor på PC
- Terskelverdi kan endres via seriellmonitor

HVORFOR DENNE LØSNINGEN?

- USART er veldig enkelt å sette opp og bruke i forhold til å bruke I2C
- BAUD-raten er mer enn rask nok for denne applikasjonen
- Ved å lage et bibliotek er det enkelt å utvide prosjektet, for eksempel med flere sensornoder snakker sammen med seriellkommunikasjon

IDEER TIL BRUK OG VIDERE UTVIKLING

- Dette er et avkjølingsprogram. Programmet overvåker både temperatur, og mekanisk endring som resultat av økt temperatur.
- Sammenhengen mellom temperaturmålingen og viftehastigheten fungerer i praksis likt som en CPU-vifte i en PC. En kan se for seg å videreutvikle/tilpasse programmet til å brukes i denne typen elektronikk.
- Programmet er laget for å lett kunne legge til flere inputkanaler i ADC-avlesningen. Det eneste som trengs er å legge til en ekstra case i Switch-case-strukturen i ISR-funksjonen. Man må i så fall også lage en annen metode for svitsjing av inputkanaler som for eksempel en for-løkke.