

FYS3240 Lab 3

Oppgave 1:

I denne oppgaven skal dere bruke den innebygde ADC'en til å måle på diverse sensorer. Xplained-kortet har innebygget temperatur- og lyssensor som er koplet til hver sin pinne av ADC-porten.

ADC'en skal konfigureres til å hente ut data fra sensorene på kortet. Det enkleste er å starte med å måle på potentiometeret eller lyssensoren, da det er lett å se en klar tilbakemelding fra disse.

Potentiometeret er koblet til pinne 1 på PORTB, så multiplexeren må rettes til denne pinnen for å måle på den. Dere kan velge selv om dere vil sette opp ADCB til differensiell eller single-ended måling. (Hint: databladet, fra s. 323)

Touchbryterne skal brukes til å velge mellom de 3 sensorene: temperatur (NTC), Potentiometer, lys (LDR).

Kommenter valgene dere tar!

Hvor de andre sensorene er koblet finner dere i databladet til utviklingskortet.

Grunnen til at denne oppgaven er svært kort er at all informasjon som trengs for å gjennomføre den finnes lett tilgjengelig i datablad.

Følgende metoder finnes i `fys3240_adcb.h`:

`void adcb_init();`

Initialiserer alle relevante registre for ADC.

Dette inkluderer å sette opp interrupts på ferdig konvertering. Dere kan bli nødt til å tenke over interrupt-prioriteten nå slik at alle interrupts for gjort sitt.

Prescaleren i ADC'en kan med fordel brukes slik at det oppnås en lavere samplingshastighet (vi trenger ikke veldig høy hastighet i denne oppgaven), og eventuelt også prescaleren til systemklokka.

`adcb_callback(void);`

Denne rutinen skal oppdatere en variabel med verdien som leses ut fra ADC'en. Variabelen den skriver til må være en volatile `uint8_t`. Denne kalles fra ISR(<adc-vektornavn>) i testapplikasjonen.

Denne rutinen skal også sjekke hvilken bryter som eventuelt er trykket og skifte kilde i forhold til det.

`lcd_for_adc_callback(void);`

Skal oppdatere verdien på skjermen til den verdien som er lest ut av ADC tidligere.

Denne kalles fra ISR(<lcd-vektornavn>) i testapplikasjonen.

Her blir dere nødt til å dele verdien gradvis ned slik at et og et siffer kan hentes ut. Dere trenger bare en enkel algoritme for dette, men dere må finne den selv.

Følgende hjelpefunksjoner skal også lages:

`show_value(uint8_t val)` for å vise en verdi på LCD-skjermen. I en fler-sifret verdi må et og et siffer plukkes ut.

`conv_ascii(uint8_t val);`

Brukes til å konvertere en integerverdi til ascii.

Vi skal ikke bry oss så mye med å skalere verdien i forhold til hvilken sensor det måles på.

Valgfritt:

Del 2:

I del 2 skal det implementeres en klokke.

Denne skal implementeres ved hjelp av enten Real-Time Counter eller Timer/Counter 0.

For å sette opp Timer/Counter type 0 til å brukes som klokke vil vil første dele klokken så mye som mulig. Start derfor med å sette prescaler til å dele klokken med 1024. (Databladet side 165)

Deretter må en verdi settes til *Compare or Capture* registeret. Timeren sammenligner løpende den opptalte verdien med denne og sender ut en interrupt når den når verdien.

Interrupt-handleren for TCO skal alltid få prioritet for å telle riktig, og eventuell bruk av sei() og cli() i ISR'en til LCD-driveren bør fjernes.