Høgskolen i Østfold

EKSAMEN Sensorveiledning / Løsningsforslag

Emnekode: IRE 14021 **Ordinær Eksamen høst 2023.**	**Emnenavn:** Anvendelse av digitalteknikk, mikrokontrollere og programmering
Dato: 13. des. 2023 **Sensurfrist:** 3. jan. 2024	**Eksamenstid:** 3 timer
Antall oppgavesider: 5 **Antall vedleggsider:** 0	**Faglærer: Reidar Nordby , Olav Aaker** Reidar tlf. 90530571 Olav tlf.94806430 **Oppgaven er kontrollert:** Ja
Hjelpemidler: *Kalkulator, med tomt minne, som ikke kan regne symbolsk eller kommunisere trådløst.* Digital fundamentals Floyd, Thomas L. ISBN: 1292075988; 9781292075983 Utgave: 11th ed., Global ed.	
Om eksamensoppgaven: Les nøye igjennom alle oppgavene og prioriter! Oppgavene dekker fire tema med varierende antall underpunkt.	
Kandidaten må selv kontrollere at oppgavesettet er fullstendig	

Høgskolen i Østfold

Mikrokontrollerarkitektur, hukommelse og datakonvertering

1) Gi en beskrivende sammenligning av Harvard og Von Neumann arkitekturene. Hva er fordelene med Harvardarkitekturen i «små» mikrokontrollere?

Harvard:	*von Neumann:*
Separat bussystem for program og data	*ETT stort bussystem*
Fast instruksjonslengde	*Fleksibelt system*
Samtidig lesing av instruksjoner og data	*Ingen parallellitet*
Innbygd beskyttelse av instruksjonsblokker Med egen hukommelse for programm er ikke denne tilgjengelig utenfra	*Kjøring av data som program er mulig Virusteknikker som «Data stack overflow» og «Data heap overflov» kan benyttes for å kjøre inntrengende kode.*

SENSUR: Det forventes at kandidaten kjenner til dobbeltbussystemet i Harvard arkitekturen.

2) Hva menes med «paged memory»?

For mikrokontrollere:
Programtelleren deles i to PCH (Pages) og PCL (Index) Pages refererer til programmer (Hovedprogram, Avbruddsrutiner og funksjoner) mens Index fungerer som programadresseteller innen hvert sitt programdel. Dette danner grunnlaget for treading.
For større CPU'er: Begrepet refererer til hentingen av datablokker fra lagringsmedia og plassere disse tilgjengelig i RAM.

SENSUR: Dette er et «A» spørsmål, informasjonen er vanskelig tilgjengelig for mikrokontrollere

3) Hvilke trinn består en AD konvertering av? Gi beskrivelse av alle operasjonene fra analogsignal til digital tallstørrelse.

1 Sampling and holding: - Som låser en analog verdi fast til et nivå fra et bestemt tidspunkt satt av samplingsklokka.

2 Kvantifisering: - Binder nivået på holdingen til et forutbestemt intervall spenningsmessig.

3 Koding: - Hvert spenningsintervall er assosiert med en kode som i sin tur tildeles den aktuelle prøven (sample). Denne koden overføres til prosessor / hukommelse.

Høgskolen i Østfold

SENSUR: Det forventes kjennskap til denne prosessen. Legg særlig vekt på tidfesting og intervall.

Stack

4a) AVR mikrokontrollerene har alle minst to stackpekere, hva benyttes minst en av dem til når du programmerer i «C»?

En stackpeker benyttes alltid til å anngi tilbakehoppadresser og mikrokontroller status.
Den andre benyttes til parameteroverføring ved funksjonskall (så lenge du programmerer i «C»

4b) Gi beskrivelse av hvordan toppen av programstack'en ser ut når en funksjon med hode som under kjører:

```
double  <operasjon> ( char Aa, int Bb, int  Cc,){

...

...

}
```

Innholdet i programstack:
De 8 Høyeste adressebit til neste instruksjon som skal kjøres etter ferdig rutine
De 8 Laveste adressebit til neste instruksjon som skal kjøres etter ferdig rutine

Innholdet i parametersteck ved starten av rutinen:
Cc 2 byte
Bb 2 byte
Aa 2byte

Innholdet ved retur fra rutinen:

Double 4 Byte

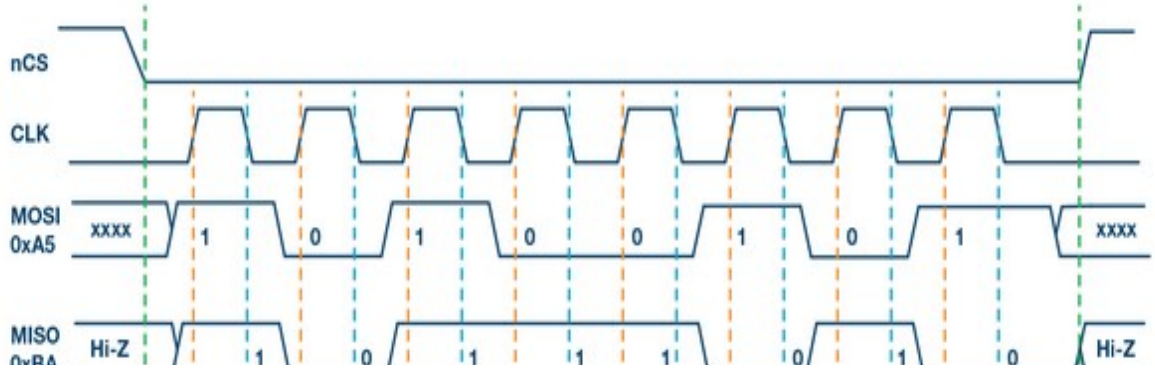
SENSUR: Dette er et vanskelig spørsmål å besvare og krever dyp innsikt i mikrokontrollerens arbeidsmåte. Merk: Det spørres bare etter programstack!

Seriekommunikasjon

Høgskolen i Østfold

Velg 1 av de neste 2 oppgavene: 5) eller 6)

Oppgave 5 velg enten denne oppgaven eller Oppgave 6



Figur hentet fra: https://www.analog.com/en/analog-dialogue/articles

5a) Hvilken form for seriekommunikasjon viser figuren?

Figuren henviser til SPI kommunikasjon

5b) Hvilken funksjon har signalet nCS? Beskriv, og kan det være flere av dem tilknyttet samme buss?

nCS står for Chip select nr n. Det finnes EN CS for hver chip koblet til bussen.

5c) Hva skjer når klokka går fra lav til høy?

Data på MOSI og MISO blir klokket ut.

5d) Og når klokka faller?

Data på MOSI og MISO er stabile og klokkes inn.

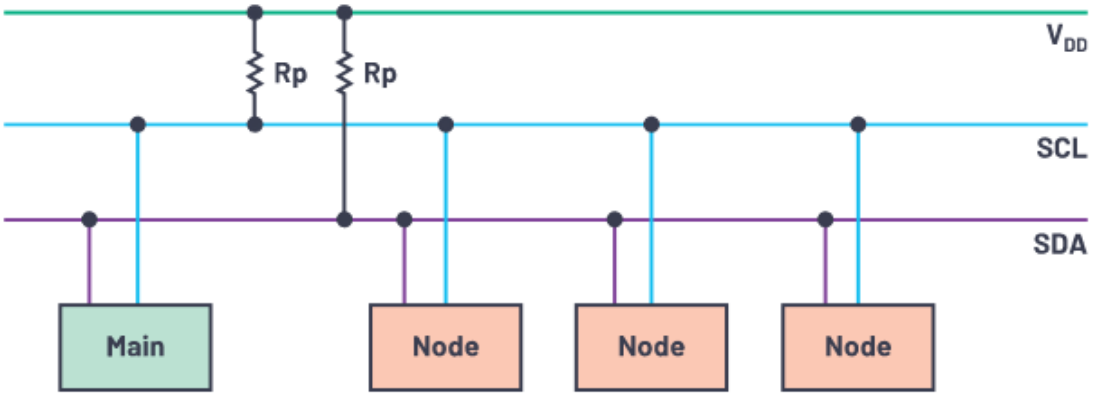
5e) Hvilken enhet (master , main eller slave, subnode) sender klokkesignalet?

Master også kalt main sender klokkesignalet.

SENSUR: Dette stoffet er ikke undervist, det er derimot krevd brukt i studentenes prosjektoppgave. Det har vært fritt valg mellom SPI og I^2C avhengig av valgte chipfunksjoner. Alle kandidatene bør kjenne forskjellene i busstrukturen.

Høgskolen i Østfold

Oppgave 6 velg enten denne oppgaven eller Oppgave 5



figur hentet fra https://www.analog.com/en/analog-dialogue/articles

6a) Hvilken form for seriekommunikasjon viser figuren?

Figuren viser I^2C kommunikasjon

6b) Hvilken funksjon har signalene SCL og SDA? Beskriv begge signalene

SCL serial clock benyttes til synkronisering av signalene mellom enhetene.
SDA Serial Data overfører bitvis informasjon mellom enhetene synkronisert med SDA.

6c) Hva skjer når klokka går fra lav til høy?

Data legges ut på SDA.

6d) Og når klokka faller?

Data klokkes inn i mottagere(ne) fra SDA

6e) Hvilken enhet (main eller node) sender klokkesignalet?

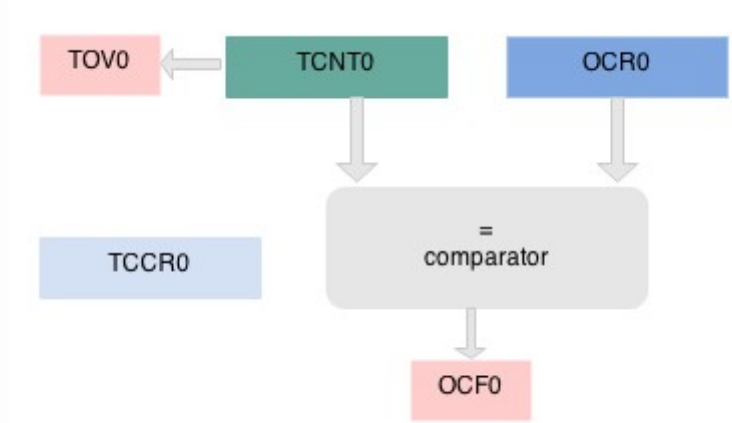
Klokkesignalet genereres av Main. (Også kalt Master)

SENSUR: Dette stoffet er ikke undervist, det er derimot krevd brukt i studentenes prosjektoppgave. Det har vært fritt valg mellom SPI og I^2C avhengig av valgte chipfunksjoner. Alle kandidatene bør kjenne forskjellene i busstrukturen.

Høgskolen i Østfold

Tidskontroll og avbrudd

Oppgave 7



figur hentet fra: https://exploreembedded.com

7a) Hvilken periferienhet (relativt til CPU) viser figuren?

Timer Counter 0 i AVR ATMEGA 8 kjernen.

7b) Flagget TOV0 Aktiveres når?

Timer Overflow 0 aktiveres når telleren når maksimal verdi (0FFh).

7c) Hva lagres i TCCR0 ?

Dette er kontrollregistret til Timer0 og holder bruksmodus og status for telleren

7d) Flagget OCF0 benyttes til ?

OCF0 benyttes til å generere avbrudd når telleren har nådd en bestemt verdi.
TOV0 benyttes også til å gi avbrudd, tilsammen kan disse to danne grunnlaget for en Puls-Bredde Modulert utgang (PWM).

7e) Beskriv formål og virkemåte til denne enheten.

Formålet med periferienheten er å kunne telle eksterne eller interne pulser (klokkepulser) og gi avbrudd når et visst antall pulser har forekommet. Sekundært kan enheten benytte til å gi et pulsbreddemodulert signal når begge flagg benyttes.
Selve enheten består av en 8 bit teller med et sammenligningsregister og overløpsflagg. Telleren teller opp på grunnlag av pulser enten fra en ekstern pinne eller fra systemklokka via en frekvensdeler (prescaler). Flagg, som kan gi avbrudd, reises etter lik verdi ved sammenligning og ved overløp. Normalt settes en ny startverdi i telleren fra ei rutine som startes på grunnlag av overløp (Polling eller avbrudd). Default er 0x00h

SENSUR: Legg vekt på forståelse av virkemåten.

Høgskolen i Østfold

8a) På neste side er er gitt to programmer med samme funksjon, forklar forskjellene.

Begge programmene får en LED koblet til PD4 til å blinke.

Det øverste programmet benytter timer 0 og hva som kalles polling ved å kontinuerlig sjekke tilstanden til Timer Overflow 0 flagget. Timer Overflow oppstår hver gang telleren TCNT0 passerer sin toppverdi ved 0xFFh.

Det nedre programmet benytter seg av timer 1 til å gi et reelt avbrudd og behandler dette i avbruddsrutinen ISR (TIMER1_OVF_vector) for å gi en presis blinkefrekvens på 1/2 Hz.

8b) Hva er fordelen med å benytte teknikken benyttet i det nedre programmet?

Fordelen er at den avbruddsdrevne prosessen kan kjøres i parallell med andre prosesser/programmer.

8c) Forkortelsen ISR står for?

Interrupt Service Routine . Avbruddstjenesterutine på norsk. Benyttes som fellesnavn på alle avbruddsrutiner. Hver avbruddskilde har sin spesielle avbruddsrutine-vektor som benyttes som argument til avbruddsfunksjonen; ISR (avbruddskilde).

8d) Hvorfor angis setningen «TCNT1 = 63974» to ganger i det nedre programmet?

Setningen benyttes første gang i hovedprogrammet (main) for å «bestille» første avbrudd om 1 sekund. Andre gang setningen benyttes er i avbruddsrutinen hvor den på-nytt setter et avbrudd om 1 sekund. OBS:Dette er modifisering av tellerverdien som automatisk settes til 0x0000h etter overløp (overflow).

8e) Kommandoen *sei()* har hvilken virkning?

Funksjonen SetEnableInterrupt (sei()) tillater avbrudd. Avbrudd er ikke tillatt før denne funksjonen opptrer.

8f) Kommenter linje for linje hva som skjer i nederste program. Få med hensikten med hver linje. Kommentarer som står hjelper deg på vei..

Kommentarer er gitt som // kommentar i kursiv i programlisting.

SENSUR: Denne oppgaven er oppgavesettets hovedutfordring. Legg vekt på forståelsen av avbrudd og bruken av den generelle interruptrutinen ISR (avbruddskilde)

Høgskolen i Østfold

```
#include<avr/io.h>
#define LED PD4

int main()
{
        uint8_t timerOverflowCount=0;
        DDRD=0xff;        //configure PORTD as output
        TCNT0=0x00;
        TCCR0 = (1<<CS00) | (1<<CS02);

       while(1)
       {
        while ((TIFR & 0x01) == 0);
         TCNT0 = 0x00;
         TIFR=0x01; //clear timer1 overflow flag
        timerOverflowCount++;
        if (timerOverflowCount>=6)
        {
                PORTD ^= (0x01 << LED);
                timerOverflowCount=0;
        }
       }
}
```

```
#include<avr/io.h>          // benytter bibliotek for inn/ ut periferienheter.
#include<avr/interrupt.h> // benytter bibliotek for avbrudd

#define LED PD4          // Gir pinnen PD4 navnet LED

ISR (TIMER1_OVF_vect)    // Timer1 ISR     //avbruddsfunksjon med kildevektor
{
       PORTD ^= (1 << LED); // Skifter verdi på pinnen PD4 (som nå heter LED)
       TCNT1 = 63974;   // for 1 sec at 16 MHz  startverdi for Timer1
}

int main()
{
       DDRD = (0x01 << LED);     //Configure the PORTD4 as output

       TCNT1 = 63974;   // for 1 sec at 16 MHz    startverdi for Timer1

       TCCR1A = 0x00; // Ingen pinner eller PWM benyttes
       TCCR1B = (1<<CS10) | (1<<CS12);;  // Timer mode with 1024 prescaler
       TIMSK = (1 << TOIE1) ;  // Velger avbruddskilde til å være Timer1 overløp
       sei();  // tillater generelt avbrudd; set enable interrupt flag

       while(1) { } // ingen aktivitet i hovedprogrammet
}
```

Slutt på eksamensoppgavene