

# Afleveringsopgave 5

## 1 OPGAVEN

Lav et program, som blinker de tre vandrette segmenter med følgende ca.hastigheder: 5 Hz for segment a, 2 Hz for segment b og 1 Hz for segment d. Alle andre segmenter skal være slukket til alle tider.

Opgaven skal løses vha. et interrupt på Timer0 ca. hvert 10. millisekund, og I skal bruge en clockfrekvens på 1 MHz. Løsningen skal udføres ud fra nedenstående pseudokode. Brug dedikerede registre til de fire tick\*variable, og vurder deres funktion.

## 2 MIN LØSNING

Først starter jeg med at definere interrupt vektoren for Timer0 og sætter den til at hoppe til min Timer0\_Interrupt rutine.

Så sætter jeg PORTB op som output da den styrer displayet.

Jeg sætter timer0 op til at køre i CTC-mode og normal operation på OC0-pin og prescaleren skal så bestemmes for at ramme de ca. 10ms. Til dette har jeg opstillet et skema der viser de forskellige prescaler værdier og OCR0 værdier plus en grundet at selve interrupt først kommer countet efter at der har været et compare.

<i>Prescaler (Clock/value)</i>	<i>OCR0+1</i>	<i>Fejl</i>
1024	10	240us
1024	9	-784us
256	39	-16us
256	40	24us
64	156	-16us
64	157	48us
8	1250	0s

Som det kan ses kommer en prescaler værdi på 256 og 64 begge ned på 16us i fejl. Og jeg har så valgt at bruge de 256. Det kan også ses at hvis man havde haft fat i en som minimum 11-bit timer så ville man kunne ramme spot on på de 10ms.

Dernæst sørger jeg for at enable output compare match interrupt på timer0 i TIMSK registret og slår globalt interrupt til.

Jeg har ikke valgt at gøre yderligere brug af subrutiner i min løsning så alt foregår bare i mit loop.

Jeg starter med at tjekke om tick er større end nul og hvis den ikke er det så looper programmet bare indtil tick er større end nul. Når den så er det så trækkes der en fra tick og toggling af de forskellige segmenter kan begynde. Jeg beskriver kun for segment A der blinker ved 5Hz, da det er præcis samme kode for dem alle på nær et par immediate værdier der loades ind.

***Tick\_5Hz\_Handling:***

- Først lægges en til tick værdien for de 5Hz
- Så tjekkes der om værdien har nået 10 endnu, da det giver et delay på 2.5Hz for et toggle som ender ud i 5Hz blinkende.
- Hvis ikke 10 er nået endnu, så gå videre til næste segment ellers så nulstil tælleren.
- Indlæs den nuværende værdi af portb i R16 og lav en kopi af det i register til R17 og komplementer det register. R16 forbliver komplementeret da segmentet sådan set så allerede er "toggled"
- Mask alt andet ud af R16 end det bit hvor segmentet sidder til
- Mask segment bittet ud af den kopi vi lavede i R17 og bibehold værdien af de to andre segmenter
- OR de to registre sammen så der kommer en maske ud hvor segment a er togglet og de andre er bibeholdt.
- Komplementer så fordi at displayet er active-low og opdater så displayet.
- Gå videre til opdatering af næste segment

Det samme gør sig gældende for opdateringen af de andre segmenter. Og til slut startes loopet forfra.