Jonas A. L. Andersen jonaa14@student.sdu.dk MIC 14/08-2015

Afleveringsopgave 2

1 OPGAVEN

At lave et program I assembly som kan aktivere segmenterne på et syv-segmentsdisplay i en bestemt rækkefølge. Den skal så konstant stå og skifte og hastigheden hvorved den skifter skal så afhænge af, hvilke switche der er aktiveret på boardet. Sekvensen skal starte i segment af og hvis ingen af switches er tændt så skal det tolkes som at displayet skal stå stille på det element det er nået til i sekvensen.

2 MIN LØSNING

Jeg har valgt at løse opgaven ved at gemme den ønskede skiftesekvens af displayet som en liste i programhukommelsen og derefter iterere igennem den indtil jeg når til det sidste led i sekvensen, hvor det hele så starter forfra. Dette gør at jeg ikke er låst til en bestemt sekvens, men nemt kan ændre den sekvens der itereres igennem og længden af denne.

Jeg startede med at løse opgaven angående debouncing. Jeg lavede en include-fil med en delay subroutine, hvor jeg kan ændre delayet med 1,007ms * x, hvor x er den værdi jeg sender med når jeg kalder macroen. Jeg satte så delayet til 1,007ms, da det i og for sig ikke er vigtig for funktionaliteten, men et krav stillet i opgaven.

Når microcontrolleren starter så kører jeg min INCREMENT_7SEG subroutine en gang for at sætte displayet til at vise det første segment. Dernæst indlæser jeg switches med min READ_SWITCH subroutine der både sørger for debouncing samt at komplementere den indlæste værdi da switches er active low. Dernæst tester jeg for om switches var 0, dvs. ikke aktiverede. Hvis de var det så startede jeg bare forfra igen og derved vil jeg ikke gå til næste segment i sekvensen. Hvis switches derimod var aktiverede på en eller anden måde, så incrementere jeg først displayet en gang og så sender jeg værdien af switches ind som et argument til min DELAY_MS subroutine. Som derefter delayer programmet i et antal milisekunder beregnet ved formlen (((SWITCH_STATE * 1000) + 7) * 1us)

Jeg har regnet lidt på, hvor lang tid det reelt tager for mit program at kører mellem hvert segment skift. Det optimale ville jo være at hvis switches var sat til f.eks. 128 så ville der tilsvarende være 128ms mellem hvert skift, men dette er svært at opnå da forskellige faktorer så som hvor lang tid switches er om at debounce også skal regnes med i det. Jeg gik så derfor heller ikke oppe i at ramme helt exact. Den maksimale delay jeg kunne beregne mig til at den ville få, dvs. ved switch værdi 255, var (256,053ms + Switch debounce) hvor switch debounce er en ukendt faktorer der kan variere alt efter, hvor lang tid switches er om at debounce. Hvis jeg regner fejl procent ud bliver det 0.41% hvilket er godt under de accepterede 5-10%. Fejl procenten vil dog stige jo kortere delay der skal være grundet at (debouncing + programtiden) er en fast størrelse der altid vil blive lagt til. Så hvis jeg regner delay ud ved switch værdi 1 så bliver delayet faktisk (2, 053ms + Debounce) hvilket fejlberegnet giver 100% dvs. det dobbelte delay, men dette kan jeg ikke rigtig gøre noget ved grundet debouncing.

Jeg har beskrevet alt koden med kommentarer i asm-filen.