

H3PD041121 Embedded II Gennemgående Opgave

- 1) Lav et program i C, der generer interrupt for hvert sekund ved brug af én af de 3 indbyggede timere i jeres ATmega processor => I skal implementere et timer interrupt der bliver aktivt hvert sekund.
- 2) Indfør en tæller der starter fra 0 ved program opstart og tæller én op for hvert timer interrupt, der har været.
- 3) Udvid jeres timer funktionalitet så i toggler med en port efter eget valg for hvert timer interrupt. Kobl en lysdiode og en modstand til den valgte port, så man kan se Lysdioden tænde/slukke ved hvert timer interrupt. I første omgang må man gerne lave denne funktionalitet inde i Timer Interrupt rutinen.
- 4) Lav nu om, så I kan tilmelde en callback funktion til jeres Timer fil. Dette skal forstås på den måde, at I ved opstart af jeres program tilmelder en callback funktion til jeres timer modul. Når der så sker et timer interrupt, skal denne callback funktion kaldes. Og det skal nu være i den kaldte callback funktion den i punkt 3 beskrevne port toggling skal finde sted.
- 5) Udvid det nu så I får den i punkt 2 beskrevne tæller først med over i jeres callback funktion. Det vil sige, at formatet for jeres Callback funktion nu skal kunne tage imod en parameter.
- 6) I jeres callback skal I stadigvæk toggle med jeres port. Herudover skal I gemme den modtagne tælle variabel i en anden variabel, som skal være tilgængelig overalt i jeres main.c fil. I jeres callback funktion skal I nu sætte et flag til true, hver gang I er blevet kaldt fra Timer Interrupt af. Dette flag skal også være tilgængelig overalt i jeres main.c fil.
- 7) I jeres main loop skal I nu teste på, om det i punkt 6 beskrevne flag er blevet sat. Er dette tilfældet, skal I sætte flaget til falsk igen og udskrive på UART værdien af tælle variablen, som I har gemt i den i punkt 6 beskrevne anden variabel. Herudover skal I også, når jeres Timer Interrupt flag er sat til true gå ud og måle på en af jeres Analoge porte. Til denne analoge port skal der være tilkoblet et potentiometer. Den digitale værdi I måler (værdi mellem 0 og 1023) vil svare til en spænding mellem 0V og 5V. Denne info skal I også skrive ud på UART'en. (vi går igennem AD konverteren ved en senere lejlighed her på klassen). Har I lyst, kan I også koble en LM35 (temperatur måler) til en af jeres analoge porte og udskrive info om den gældende rumtemperatur også.
- 8) Udvid nu jeres program, så I implementerer UART Interrupt funktionalitet. I skal bruge denne UART Interrupt funktionalitet til at håndtere modtagne karakterer fra PC/Hercules, således at små bogstaver bliver lavet om til store bogstaver og omvendt. Et kig i en ASCII tabel vil være til stor hjælp her. De modtagne (og måske) konverterede bogstaver skal sendes ud på UART'en igen.
- 9) Udvid nu igen jeres program således at I kan melde flere Callback funktioner (maks 3) til jeres timer fil. I hver callback funktion skal I toggle med en port efter eget valg. Til hver af de valgte porte skal I koble en lysdiode og en modstand.

10) Udvid igen jeres program, så I fra Hercules/PC af kan tilmelde/afmelde hver af de maks 3 beskrevne callback funktioner i punkt 9.

Dette kan gøres efter denne simple protokol:

1: Frameld Callback funktion 1

2: Tilmeld Callback funktion 1

3: Frameld Callback funktion 2

4: Tilmeld Callback funktion 2

5: Frameld Callback funktion 3

6: Tilmeld Callback funktion 3

11) Har I tid til overs kan I selv finde på flere fede features at implementere.

Opgaven beskrevet herover kan I første omgang se meget svær ud. Men husk på at mange af funktionaliteterne kan I implementere ved at kigge jer omkring inde i Moodle læringsrummet.

For at gøre opgaven lettere og lettere udvidelsesbar er det en god ide at tænke i Arrays på de steder, hvor det giver mening. Jeg kan i hvert fald se et par steder, hvor dette kunne være tilfældet.

God kodning !!!