

Afleveringsopgave 5

1 OPGAVEN

Programmer en "echo-server" på kitted, som modtager ASCII-karakterer gennem seriel-interface, og som reagerer ved at sende karakterer tilbage på følgende måde:

- a-z bliver kapitaliseret og sendt tilbage.
- A-Z konverteres til små bogstaver
- Alle andre karakteres sendes uændret tilbage

Krav udover:

- 8 data bits
- Ingen paritet
- 1 stop bit
- 9600 baud rate
- 1MHz clock frekvens

2 MIN LØSNING

Først starter jeg med at definere interrupt vektoren for modtager USART kommunikation og sætter den til at hoppe til min USART_Received rutine.

Så sætter jeg baud raten til 9600 ved at sætte U2X-bittet som så halverer baud rate divideren og essentielt gør at man kan generere den dobbelte baudrate. Grunden til at jeg gør dette er fordi at hvis man slår op i tabellen fra datasheetet over de mest gængse baudrates ved forskellige frekvenser, og hvordan disse genereres. Så ser det ud som følger:

Baud Rate(bps)	Fosc = 1.0000 MHz			
	U2X = 0		U2X = 1	
	UBBR	Error	UBBR	ERROR
9600	6	-7.0%	12	0.2%

Ud fra dette kan det ses at hvis man sætter UBRR til 6 og lod være med at sætte U2X registret så ville man generere en baud rate på 9600 med en fejl på -7% ! Hvis man derimod sætter UBRR til 12 og sætter U2X så vil man kun have en fejl på 0.2%.

Jeg sætter også USART-modulet op til at kører efter de fra opgaven specificerede antal stop bits osv. Jeg har valgt at gøre brug af en interrupt baseret løsning så når jeg enabler modulet så slår jeg også lige receive interruptet til.

Mit main loop står og gør absolut ingenting andet end at loope. Alt koden foregår i interruptet. Jeg har en enkelt funktion ud over interrupt funktionen, der sørger for at sende en byte afsted over USART.

Min interrupt funktion, USART_Received, gør følgende:

- *Indlæser den modtagede byte*
- *Checker om værdien er lavere end 'a'(97)*
- *Hvis den var det så hopper den ned og checker om værdien så er lavere end 'A'(65)*
- *Hvis den også var det så var det vi modtagede ikke et bogstav og så sendes byten blot tilbage igen.*
- *Hvis den derimod var større end 'a'(97) så tjekkes der om værdien så er større end 'z'(122)*
- *Hvis den er det så var det ikke et bogstav, hvis den ikke var større end så må vi have med et bogstav a-z at gøre. For at sende et kapitaliseret bogstav tilbage så trækkes 0x20 fra værdien og den sendes tilbage, nu som et kapitaliseret bogstav da ASCII tabellen er lavet sådan så at 'A' og 'a' ligger 32=0x20 pladser fra hinanden.*
- *Noget tilsvarende sker for et stort bogstav, her lægges blot 0x20 til i stedet for.*

Så for at opsummerer så står microcontrolleren og laver ingenting lige indtil at der bliver aktiveret et interrupt når der modtages en byte i USART receive buffer. Dette interrupt sørger så for at kode den modtagede værdi ud fra det som er blevet stillet i opgaven og sender derefter byten tilbage til afsenderen.

