

## **Formål**

Formålet med denne øvelse er at skrive en driver for Mega2560's UART.

Driveren vil senere kunne anvendes i projekter, hvor der ønskes seriel kommunikation mellem eksempelvis en PC og et Arduino Mega2560 kit.

## **Materiale**

Lektion 17: "Seriel kommunikation".

Lærebogen, kapitel 11 "AVR serial port programming in assembly and C".

"Det blå hæfte" = "Mega2660 I/O registers": Siderne 51 til 57.

## **Øvelsen**

I denne øvelse vil vi:

- Installere Atmel Studio's terminalprogram og/eller terminalprogrammet "Tera Terminal".
- Skrive og teste en driver for Mega2560's UART 0.

### Del 1: Installering af terminalprogram på PC'en.

Vi vil i denne øvelse skrive en driver for Mega2560's UART 0.

I forbindelse med test af driveren ønsker vi at kunne kommunikere med en PC.

Derfor skal der på PC'en installeres et terminal-program. Der findes mange gratis terminal-programmer, der kan downloades fra internettet.

Det terminalprogram, som vores lærebog anvender, hedder "Tera Terminal".

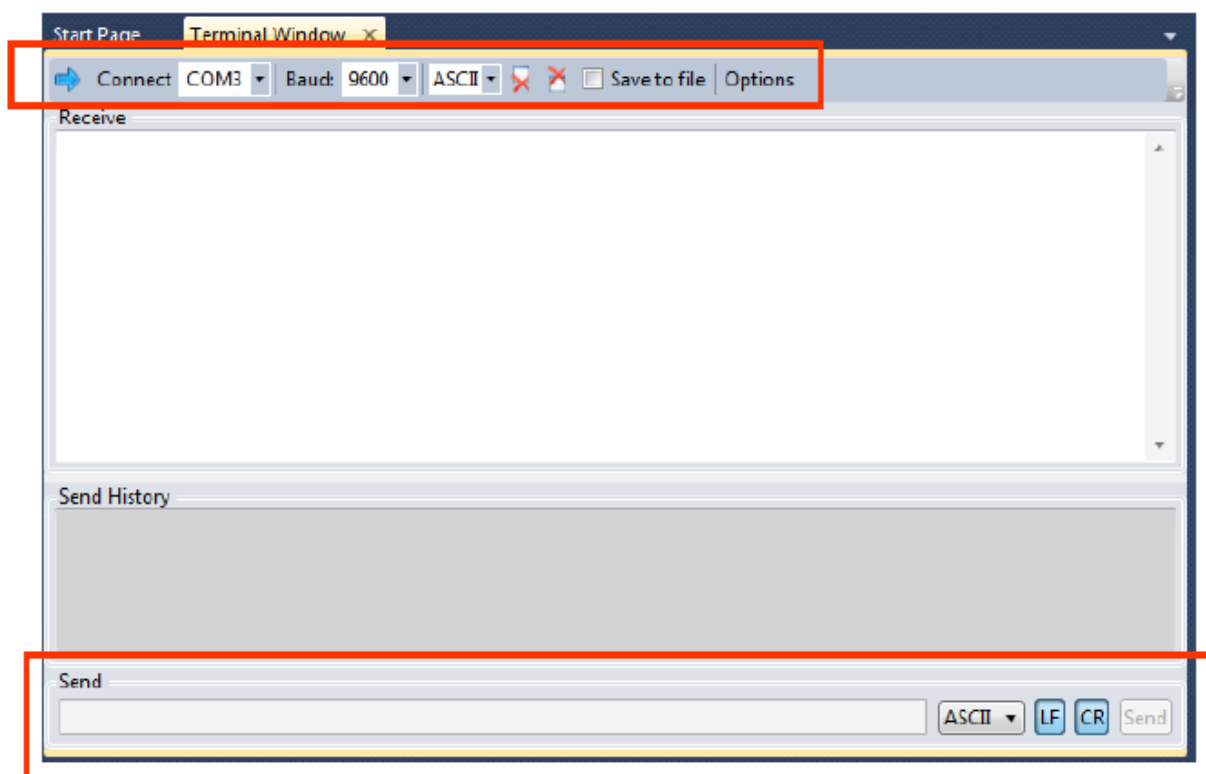
I stedet kan vi udvide Atmel Studio, så denne kommer til at indeholde et indbygget terminalprogram.

Du kan vælge at foretage begge installeringer, men hvis kun et terminalprogram installeres, anbefales Atmel Studios terminalprogram (del 1a).

### Del 1a: "Udvidelse af Atmel Studio med terminalprogram"

Du har sandsynligvis allerede udvidet Atmel Studio med terminalprogrammet som en del af installeringen (LAB øvelse 1). Hvis du ikke har menupunktet "View" -> "Terminal Window", så kørs filen "as-extension-terminalwindow-6.0-release-2.2.14-win32.win32.x86.vsix", som findes på MSYS Blackboard.

Terminalprogrammet startes som nævnt via menupunktet "View" -> "Terminal Window":



Husk før brug at klikke på "Connect". Først herefter er terminal aktiv.

Note 1: I det øjeblik man klikker "Connect", sker der et RESET af Mega2560.

Note 2: For at sende, skrives et eller flere tegn i "Send"-feltet, hvorefter der klikkes på "Send".

Note 3: Man kan vælge "LF" henholdsvis "CR" til og fra ved at klikke på dem.

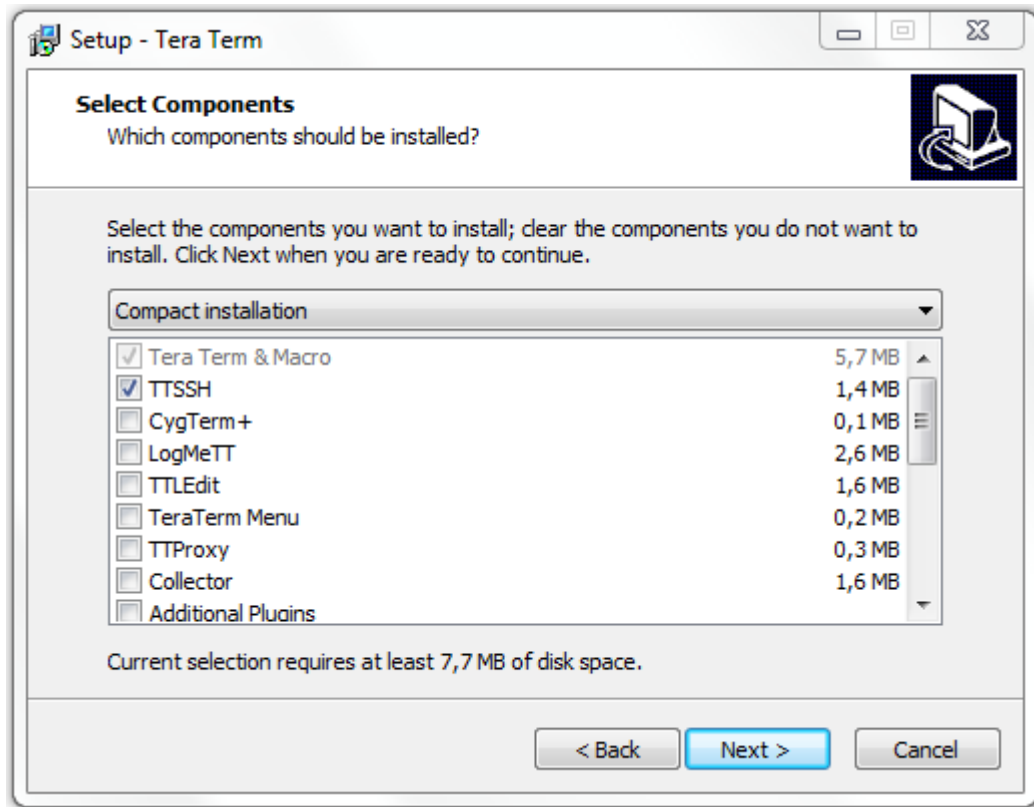
Når de er markerede, sendes også tegnene '\n' og/eller '\r' (efter det, som står i "Send"-feltet).

Normalt vil vi kun sende et tegn (det vi indtaster i "Send"-feltet), så klik på "LF" og "LF", så de ikke er blå.

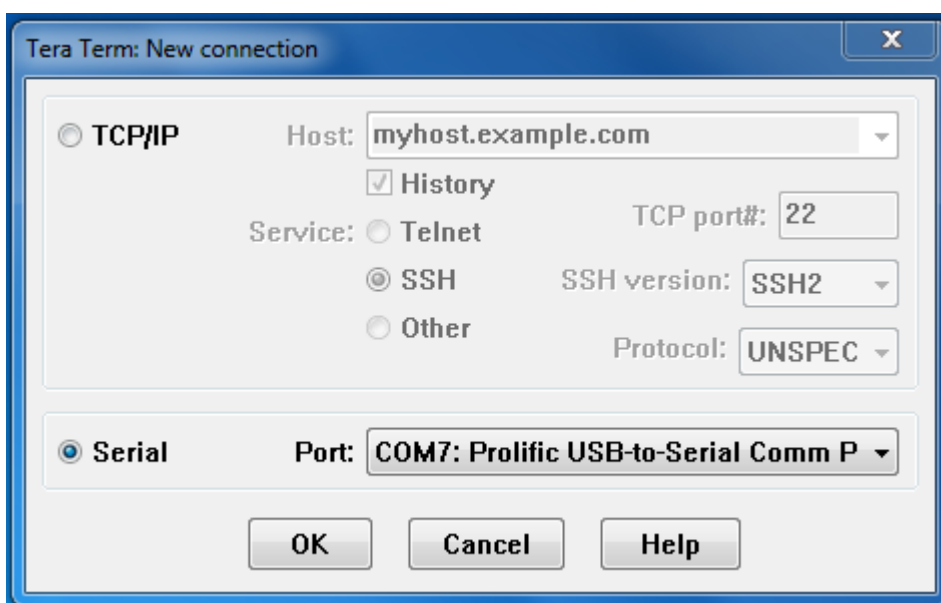
**Del 1b: "(Eventuel) installering af Tera Terminal"**

Installer "Tera Terminal" ved at køre setup-filen "teraterm-4-69.exe", der er tilgængelig på MSYS Blackboard.

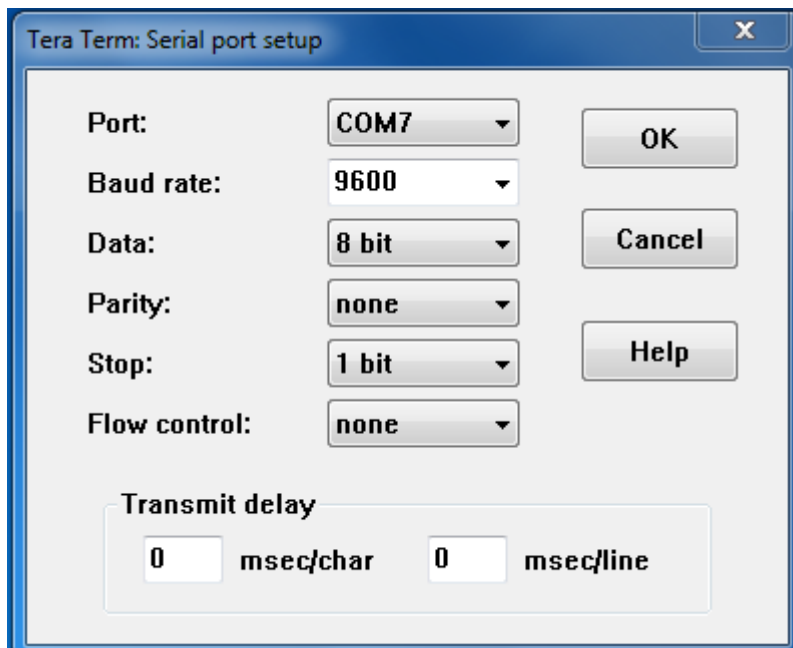
Under installeringen bør du vælge "Compact Installation" som vist nedenfor:



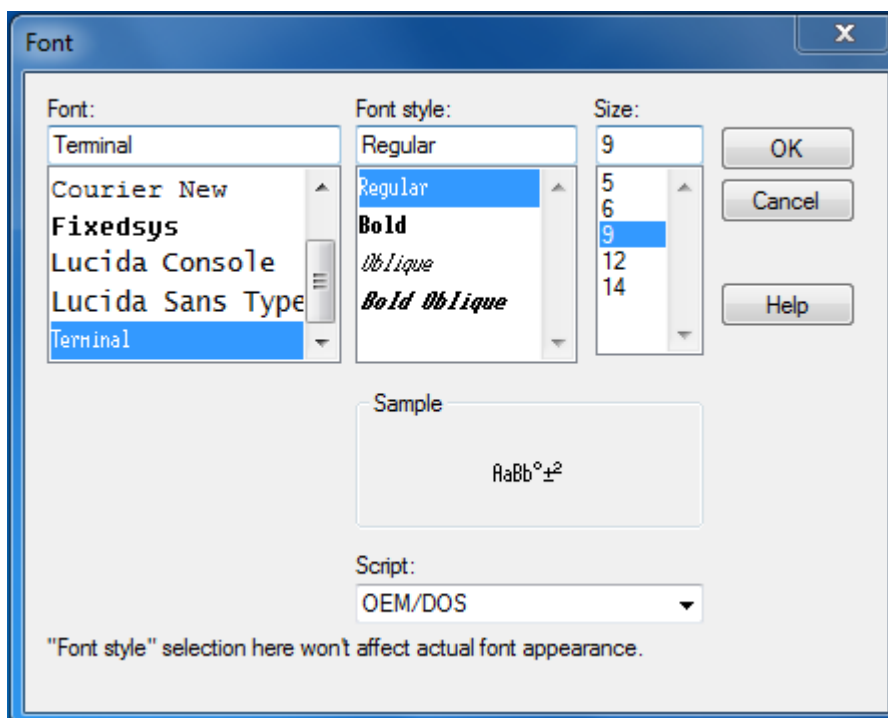
Når du herefter bruger terminal-programmet, skal du i det vindue, der straks kommer frem, vælge "Serial" (og den COM-ports, som din PC anvender) og klikke OK.



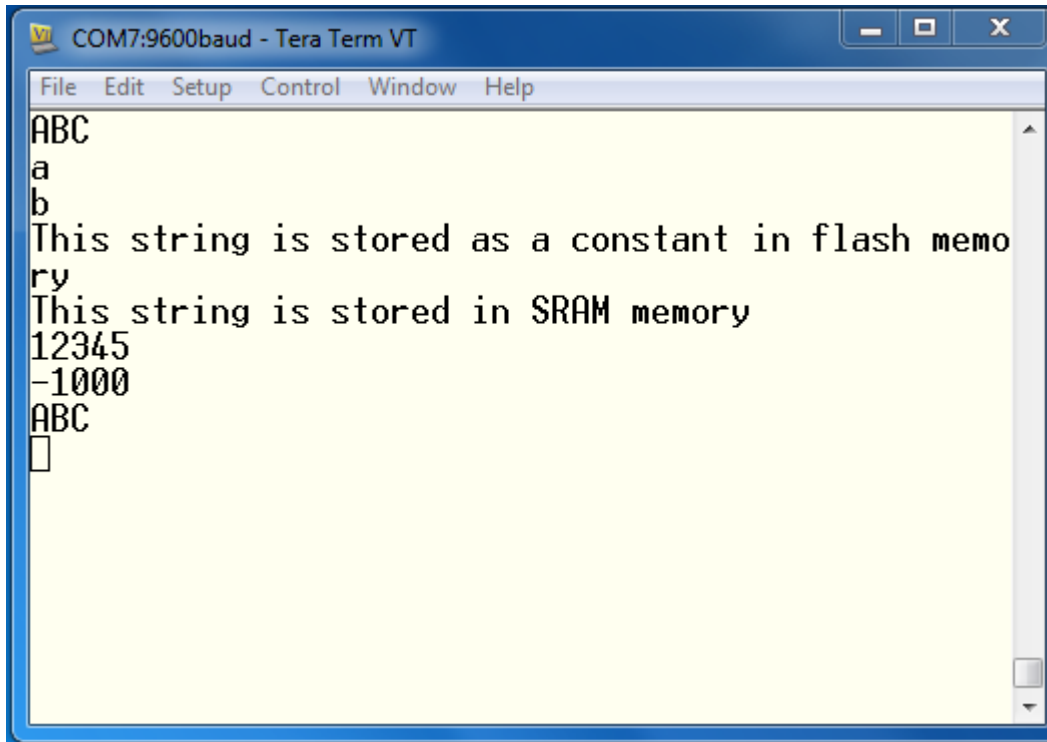
Under "Setup" -> "Serial Port" foretages indstillingerne for terminalen:



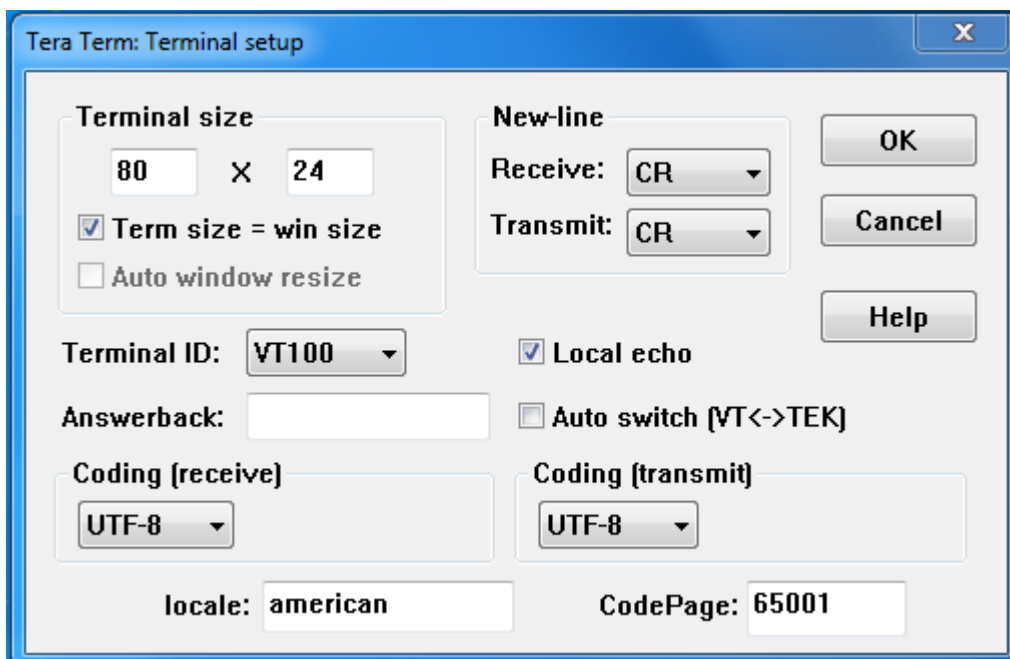
Under "Setup" -> "Font" har man mulighed for at vælge skrifttype og størrelse (hvis man ikke vil bruge default Font'en):



Ellers virker terminalen som et vindue, hvor tegn modtaget på COM-porten vises. Når man taster et tegn på tastaturet, sendes det på COM-porten.



Hvis man ønsker, at de tegn, som man taster på tastaturet også skal vises i terminalvinduet, vælges "Setup" -> "Terminal" og fluebenet "Local Echo" sættes.



Der er mange andre instillinger, som man kan lave i terminalen, men de vil ikke blive omtalt her. Du kan læse mere om dem i programmets "Online Help".

### Del 3: "UART driver"

I øvelsen skal der implementeres en asynkron driver for Mega2560's UART 0.  
Der anvendes ikke interrupts.

Driveren implementeres i en særskilt fil "uart.c" og grænsefladen beskrives via nedenstående header-fil "uart.h", der er tilgængelig på MSYS Blackboard:

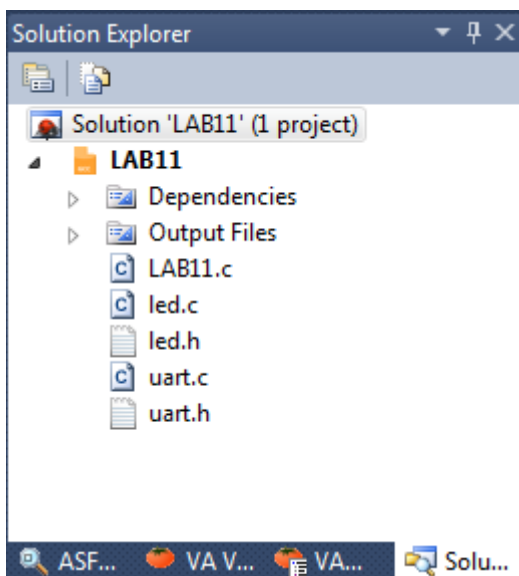
```
/*
 * "uart.h":
 * Header file for Mega2560 UART driver.
 * Using UART 0.
 * Henning Hargaard, 11/11 2015
 */

void InitUART(unsigned long BaudRate, unsigned char DataBit, char Parity);
unsigned char CharReady();
char ReadChar();
void SendChar(char Tegn);
void SendString(char* Streng);
void SendInteger(int Tal);
*/
```

Desuden findes på MSYS Blackboard et forslag til testprogram ("LAB11.c") og en skabelon for filen "uart.c".

Start derfor med at oprette et nyt C projekt (der skal hedde LAB11), som består af filerne "LAB11.c", "uart.h" og "uart.c".

Desuden tilføjes til projektet din LED port driver ("led.c" og "led.h") fra LAB7:



Opgaven går nu ud på at færdigimplementere og teste "uart.c" ved hjælp af testprogrammet.

*Beskrivelse af driverens funktioner:*

**void InitUART(unsigned long BaudRate, unsigned char DataBit, char Parity)**

Skal initiere UART 0 til den ønskede BAUD-rate (300 - 115200), det ønskede antal databits (5 - 8) og ønsket paritet ('E' = Even Parity, 'O' = Odd Parity – ellers No Parity).

Hvis Parameteren BaudRate er mindre end 300 eller større end 115200, må der ikke ske nogen initiering af UART'en.

Hvis Parameteren DataBit er mindre end 5 eller større end 8, må der ikke ske nogen initiering af UART'en.

Vi antager, at Mega2560's clockfrekvens er 16 MHz.

Den værdi, der skal skrives til UBRR0, skal i funktionen beregnes på basis af "BaudRate"-parameteren og Mega2560's CPU clockfrekvens (afrunding kan forekomme).

Desuden skal UART'en initieres til:

- Asynkron mode.
- Både RX og TX enabled.
- 1 stop bit.
- Alle interrupts disabled.

**unsigned char CharReady( )**

Meddeler, om UART 0 har modtaget et tegn.

Hvis et tegn er modtaget, returneres en værdi forskellig fra 0 ( = TRUE).

Hvis der ikke er modtaget et tegn, returneres værdien 0 ( = FALSE).

Funktionen skal ikke afvente modtagelse af et tegn, men blot returnere oplysningen om, hvorvidt et tegn er modtaget.

**char ReadChar( )**

Returnerer et modtaget tegn fra UART 0's modtageregister (UDR).

Funktionen skal først afvente, at et tegn modtages (bit RXC0 i registeret UCSRA0).

Derefter skal tegnet i UDR0 returneres.

**void SendChar(char Tegn)**

Sender et tegn via UART 0. Tegnets overføres som parameter.

Inden tegnet skrives til data registeret (UDR0), skal funktionen afvente "UART data register empty" (bit UDRE0 i registeret UCSRA0).

### **void SendString(char\* Streng)**

Udskriver en 0-termineret tekststreng ved hjælp af UART 0.

Funktion modtager som parameter en pointer til den streng, som vi ønsker udskrevet.

Pointeren peger altid på det første tegn i strengen, som altså er 0-termineret.

Brugeren har på forhånd (altså inden denne funktion kaldes) oprettet og lagret strengen.

Nedenstående viser i pseudo-kode, hvordan funktionen kan implementeres:

```
while ("Det som pointeren peger på" ikke er 0)
{
    SendChar("Det som pointeren peger på");
    Flyt pointeren en plads frem;
}
```

### **void SendInteger(int Tal)**

Denne funktion skal udskrive værdien af integer "Tal", der modtages som parameter.

Hvis man f.eks. kalder funktionen på følgende måde:

```
SendInteger(147);
```

skal følgende tegn sendes via UART 0: '1', '4' og '7'.

*Hint:*

Opret først i funktionen et lokalt array af "passende" størrelse.

Brug dernæst standard-funktionen **itoa( )** til at konvertere "Tal" til en streng, der gemmes i dette array. Husk **#include <stdlib.h>**.

**itoa( tal, array, 10 )** gemmer strengen svarende til "tal" i "array" (og 0-terminerer denne).

Brug derefter funktionen **SendString()** til at sende strengen.

Test driveren ved hjælp af testprogrammet og et terminalprogram kørende på PC'en.

Vigtigt: Man kan ikke programmere (downloade sit program fra Atmel Studio) via COM-porten, hvis terminalprogrammet samtidigt er i brug.

Derfor skal terminalprogrammet lukkes ned / disconnect'es under programmeringen.

### **Ekstraopgaver (hvis du har tid):**

1. Implementer en SW UART transmitter, der anvender et vilkårligt portben som TX terminal.
2. For "nørden":
  - Implementer en SW UART receiver, der anvender et vilkårligt portben som RX ben.
  - Der kan med fordel anvendes et eksternt interruptben.
  - Bemærk, at det er mere kompliceret at implementere en receiver end en transmitter.