

0.1 Introduktion

I denne opgave skal vi undersøge, hvordan man med en I2C master kan snakke med temperatur sensoren LM75. Hertil skal vi bruge viden fra tidligere opgaver og snakke med vores master gennem en UART forbindelse. Dette gør vi for at aflæse LM75'ens målinger.

0.2 Overvejelser

For at vi kan snakke med LM75'eren er der 2 forbindelser vi skal fokusere på.

1. LM75 \iff PSoC (I2C Master)
2. PSoC \iff PC (UART)

Disse forbindelse vil vi følgende beskrive og kigge nærmere på, hvilke udfordringer der opstår og hvordan vi planlægger at overkomme dem.

0.2.1 LM75 \iff PSoC (I2C Master)

Denne forbindelse foregår via I2C og her skal vi fra PSoC'en sende beskeder som LM75'en modtager og håndterer. Den opbygning vi skal give beskederne kan vi se i datasheetet¹. Først skal vi sætte den adresse, som vi skal kommunikere med LM75'en igennem. Dette gør vi med følgende besked struktur.

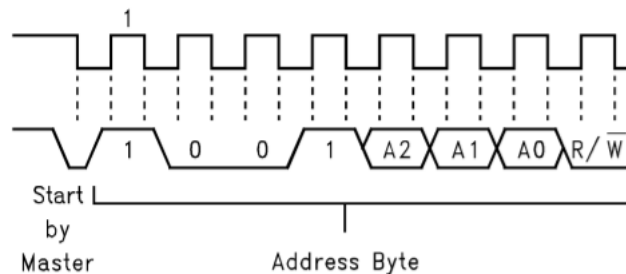


Figure 1: I2C adresse til LM75

Efter at have sat adressen skal vi skal vi modtage temperaturen. Den modtager vi som to 8-bit integers og den følger følgende format:

Temperature	Digital Output	
	Binary	Hex
125°C	0 1111 1010	0FAh
25°C	0 0011 0010	032h
0.5°C	0 0000 0001	001h
0°C	0 0000 0000	000h
-0.5°C	1 1111 1111	1FFh
-25°C	1 1100 1110	1CEh
-55°C	1 1001 0010	192h

Figure 2: Temperatur formattet fra LM75

¹<https://www.ti.com/lit/ds/symlink/lm75b.pdf>

Problematikken kommer i at få rykket rundt og behandlet de 2 bytes vi får til kun 1 enkelt byte, hvor vores temperatur er ændret fra 2's komplement til unsigned. De to bytes vi modtager kommer nogenlunde til at ligne følgende:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSB								LSB	x	x	x	x	x	x	x

I ovenstående tabel er x brugt til at vise bits som er ligegyldige for os.

Herefter planlægger vi at omskrive det til 1 byte med følgende trin:

- Gem fortegn (+/-) MSB
- Bitshift LSB til plads 15
- Bitshift MSB'en ud og resten af første byte 1 til plads 0
- OR de 2 bytes sammen så vi får LSB ind på plads 7
- Hvis MSB er 1 (-) skal vi invertere plads 0-7 og trække 1 fra for at fjerne 2's komplement
- Returner det halve og cast til en float

Da LM75'en giver os antallet af halve grader halverer vi resultatet og returner det i stedet. Så ved stue temperatur ville man få 40 fra LM75'en i stedet for 20. Dette kan vi herefter sende videre til vores PC gennem UART.

0.2.2 PSoC \iff PC (UART)

For at snakke mellem PSoC'en og vores computer bruger vi et UART komponent. Dette skal sættes op sådan at PSoC'en sender den læste data fra LM75'en til PC'en. På grund af vi ikke bruger computerens input til noget, har vi ikke noget interrupt på RX benet.

Vores computer sættes op med RealTerm til at modtage fra den USB port som PSoC'en er sat til. Selve formateringen af teksten foregår alt sammen på PSoC'en.

Et af problemer i denne forbindelse er, hvordan vi håndtere at sende vores temperatur værdi som en "floating point" værdi. Dette problem overkommes dog forholdvist nemt ved at følge en guide² givet i undervisningen. Først skal man ind i build settings og sætte float formatting til *TRUE* og herefter skal man bare øge heap sizen til *0x200*. Når dette er gjort kan man caste sin unsigned integer til en float og printe den med printf ved brug af formaterings type fieldet "%f".

0.3 Implementering

[TEXT HERE]

²GFV Lektion 5 Communication buses - lab experiment Handouts: PSoC-Creator-Printing-Floating-Point.pdf

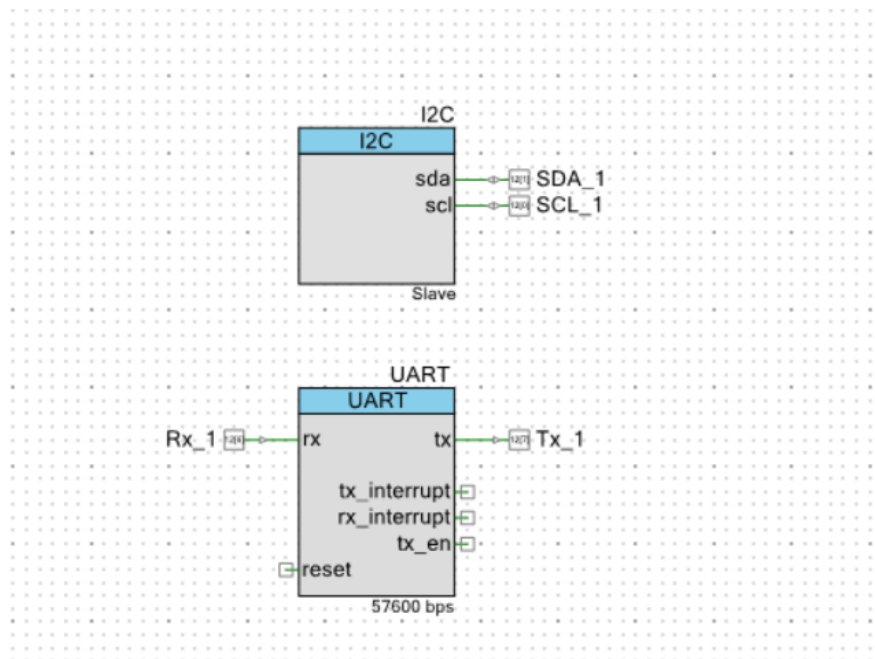


Figure 3: EN TEKST HER TAK

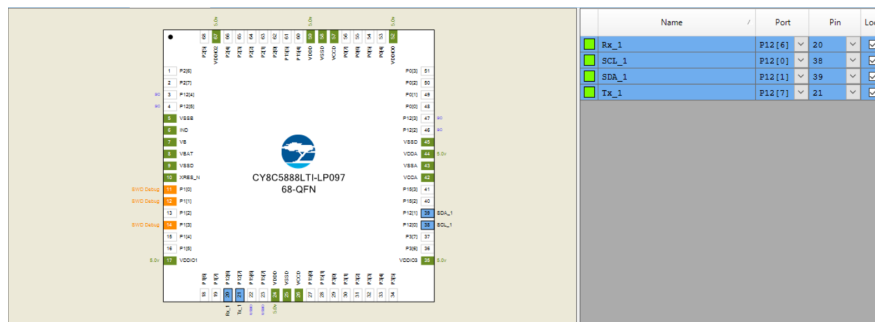


Figure 4: ENDNU MERE TEKST HER TAK

0.4 Dokumentation

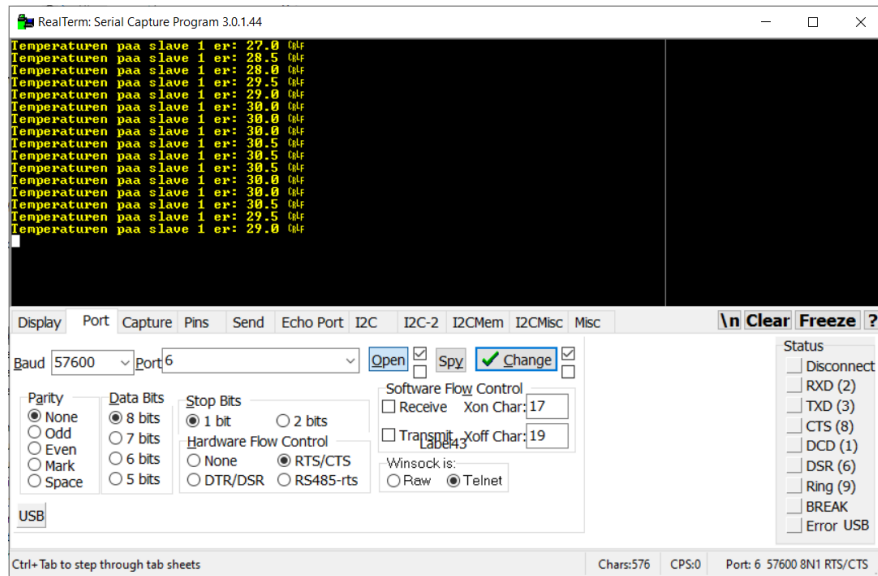


Figure 5: DER SKAL VÆRE TEKSTST

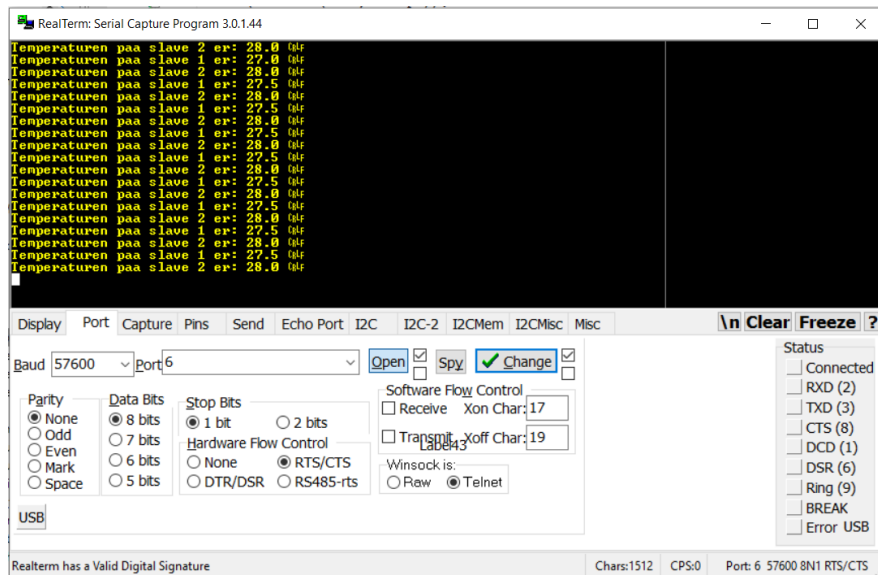


Figure 6: DER SKAL MERE TEKST SE

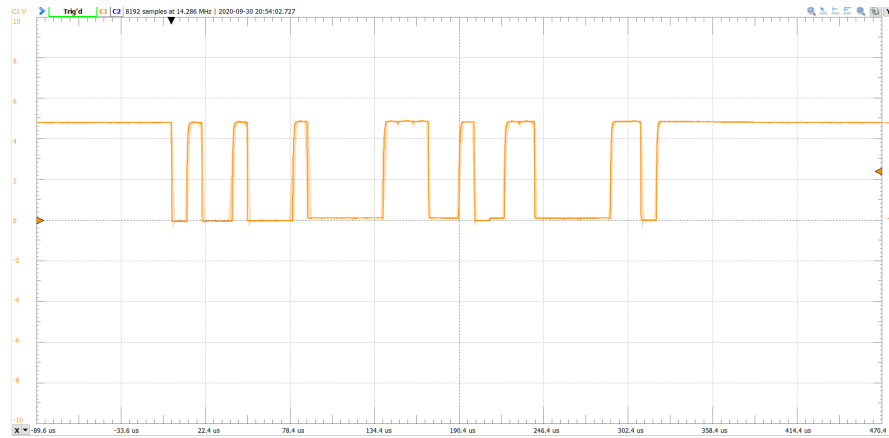


Figure 7: DER SKAL TEKST IGEN

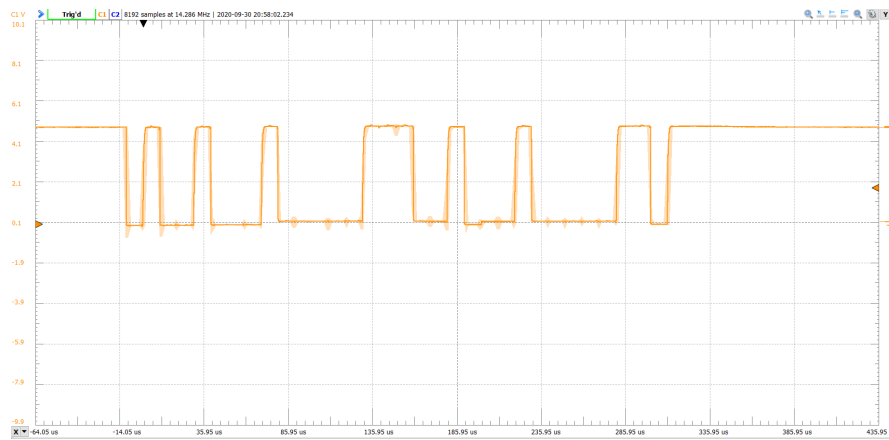


Figure 8: ENDU MERE TEKST

0.5 Diskussion

[TEXT HERE]

0.6 Konklusion

[TEXT HERE]