

## 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 datasheetet<sup>1</sup>. Først skal vi sætte den adresse, som vi skal kommunikere med LM75'en igennem. Dette gør vi med følgende besked struktur.

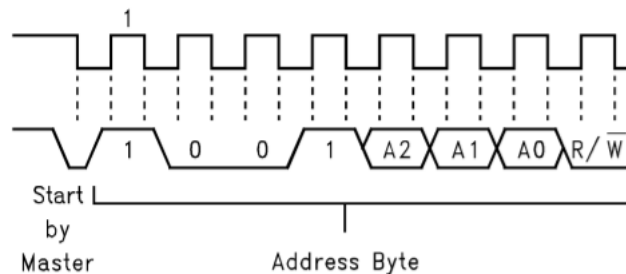


Figure 1: I2C adresse til LM75

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

<sup>1</sup><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 compliment til unsigned. De to bytes vi modtager kommer 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 compliment
- 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.

Når denne protokol er opbygget kan vi bruge den til opsætningen af flere LM75'er. Her skal vi bare indstille adressen til en anden og så kan vi forbinde dem alle serielt. Ved at gøre dette kan vi få flere LM75 slaver på samme kommunikations bus. I koden skal vi loope gennem de forskellige adresser og spørge dem en af gangen, når vi så har fået svar skal vi lukke forbindelsen og spørge den næste.

### 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.

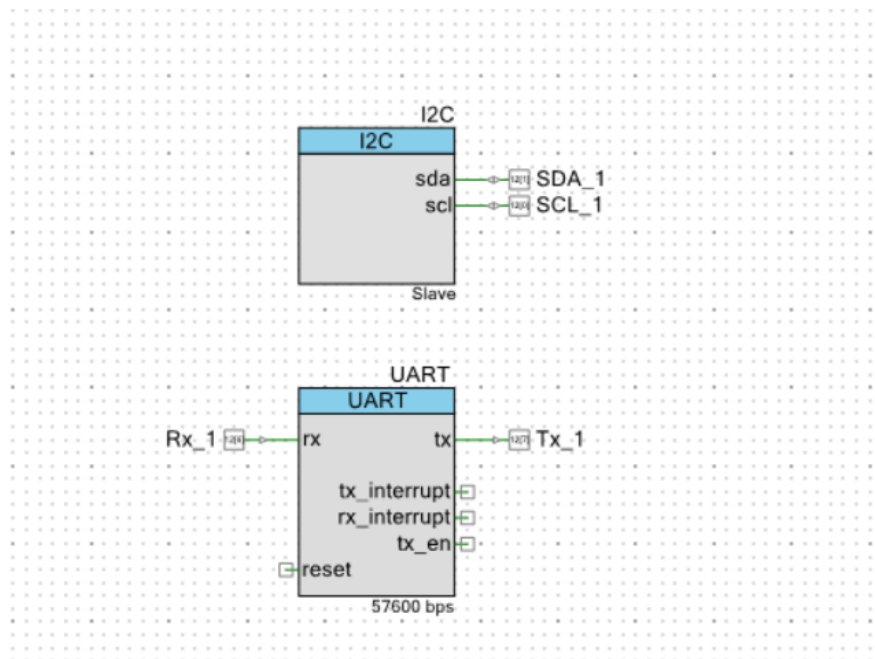
Vores computer sættes op med RealTerm til at modtage fra den USB port som PSoC'en er sat til. Selve formateringen af teksten foregår alt sammen på PSoC'en.

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 guide<sup>2</sup> 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 interger til en float og printe den med printf ved brug af formaterings type fieldet "%f".

## 0.3 Implementering

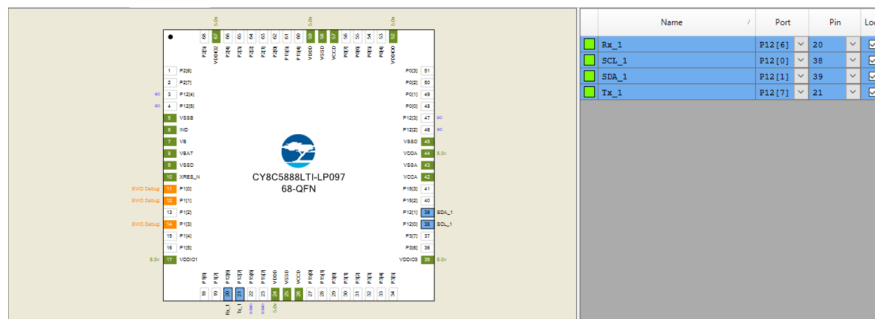
På Figure 3 kan man se topdesignet for vores PSoC. Her har vi et I2C modul til at snakke med LM75'eren og UART modulet til at snakke med computeren.

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



**Figure 3:** Top design til I2C master

Herefter bestemmer vi, hvor alle pins skal placeres. På Figure 4 kan man se, hvordan pin'sne er blevet fordelt.



**Figure 4:** Pin setup på I2C master

I koden er der 2 primære dele:

1. Indlæsning af data fra I2C
2. Print af float over UART

Disse dele vil følgende blive gennemgået og der vil blive beskrevet, hvordan koden modtager og håndterer vores input.

### 0.3.1 Indlæsning af data fra I2C

Efter at vi i `main()` loopet har startet og slået interrupten for I2C til, skal vi have lavet en funktion som modtager input fra en I2C adresse og sender det ud som en float. Vores prototype hedder `uint8 i2cRead(uint8 addr, float *result)`. Først har vi returtypen `uint8`, den bruges til at vise om vi har fået en fejl eller ej, hvis der ikke er en fejl returnerer vi '1'. Herefter har vi adressen til vores I2C device, den specificerer hvilket device vi gerne vil have data fra. Til sidst har vi resultatet. Dette modtager vi som en adresse for at kunne parse det ud af funktionen samtidigt med at vi har en fejlkode fra returværdien.

### 0.3.2 Print af float over UART

[KODE AFSNIT]

## 0.4 Dokumentation

På Figure 5 kan man se vores resultat fra computeren når vi har 1 slave på vores kommunikations bus. Her aflæser vi en temperatur fra 27 grader til 30 grader.

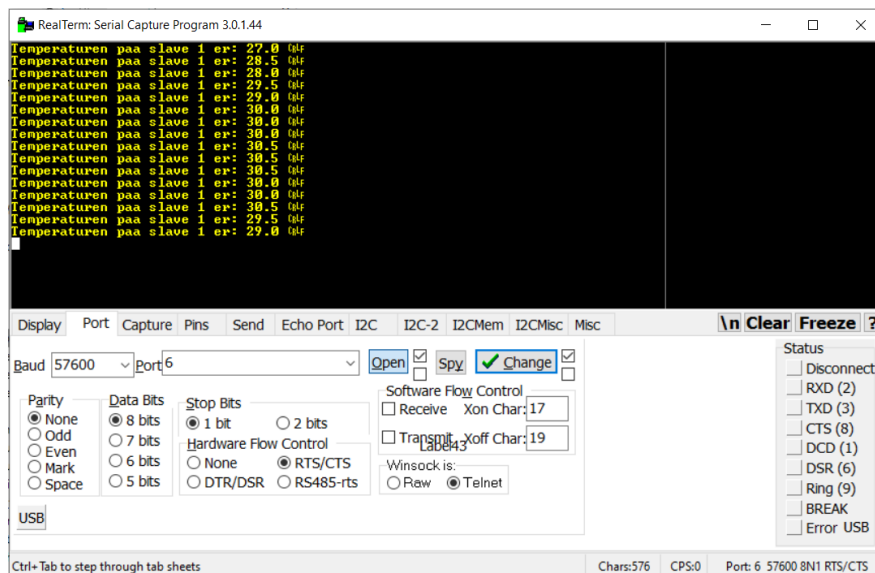
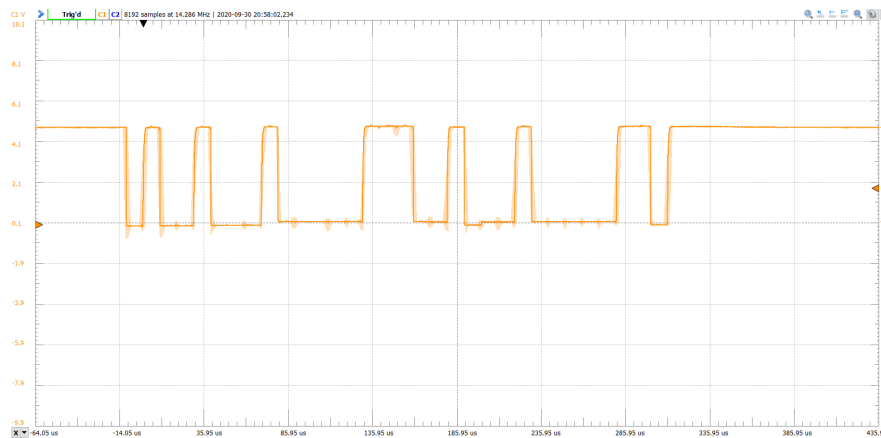


Figure 5: I2C forbindelse med 1 slave

Herefter på Figure 6 kan man se, resultatet efter vi forbandte 2 slaver til kommunikations bussen.





**Figure 8:** ENDU MERE TEKST

## 0.5 Diskussion

På Figure 5 kan man se, at temperaturen går mellem 27 grader og 30 grader. Denne ændring kom da vi placerede en finger på sensoren og det passer dermed at temperaturen stiger til 30 grader. Dette passer med forventningen om, at den omkringværende luft er koldere en vores fingrer.

Tilgængæld på Figure 6 kan man ikke se en særlig stor forskel. Denne forskel kommer af, at vi ikke placerede en finger eller et varmt objekt på en af sensorne. Dermed har vi 2 værdier som ligger meget tæt op af hinanden fordi luften omkring dem er ens temperatur.

[Diskuter waveform billeder]

[Diskuter kode ?]

## 0.6 Konklusion

På baggrund af denne øvelse kan vi konkludere, at man kan kommunikere LM75 med over en I2C forbindelse. Hertil kan man forbinde flere LM75'ere (slaver) på samme kommunikations bus og derved aflæse fra flere slaver over samme forbindelse. Herudover kan vi konkludere, at vi modtager temperaturen over 2 Bytes og med forholdsvis få trin kan man få det omregnet til en enkelt byte, som man herefter meget nemmere kan sende til sin computer og få vist.