

# Test af BMP280 med I2C

På en Raspberry Pi

Foto kommer ... engang

**Forudsætninger:** Adgang til BMP280 manualen, et godt kendskab til C programmering, og håndtering på chip (register) niveau.

**Oversigt:** Jeg beskriver hvordan man tilgår enheden fra kommandolinien (CLI) og viser derefter med et par C programmer hvordan man kan trække data ud af enheden, til brug i et C program.

Denne og andre kan findes på: <https://synkro.dk/bog>

DB: Henrik Kressner

Dec 2022

# Table of Contents

- 1. Indledning.....2
- 2 Test fra CLI.....3
- 3. Læs et register.....5

# 1. Indledning

Dette er mine foreløbige erfaringer med et BMP280 baseret barometrisk/temperatur modul tilsluttet en RPi.

Mit modul er fra AZDelivery og indeholder IKKE den originale BMP280 fra Boch, men den ser ud til at virke samme måde, måske bare ikke helt så præcis.

Jeg er naturligvis inspireret af den officielle kode på github.

## 2 Test fra CLI

Start med at sikre i2c tools er installeret, på en RPi gøres det således:

```
# apt install i2c-tools
```

Derefter kan man tjekke om der er et i2c modul på bussen, ved at skrive:

```
# i2cdetect -y 1
```

Svaret på min rpi, med et BMP280 modul tisluttet i2c ben 1,3,5 og 9 på GPIO.

```
# i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- 76 --
```

Her fremgår det at der er et device på adressen 0x76. Det passer med oplysningerne fra manualen.

Dataindholdet af den enhed kan tjekkes med:

```
# i2cdump -y 1 0x76
```

På min RPi med et BMP280 modul tilsluttet ser det således ud:

```
# i2cdump -y 1 0x76
No size specified (using byte-data access)
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  0123456789abcdef
00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
80: 93 50 89 70 64 61 99 06 19 6e e8 67 32 00 3a 92  ?P?pda???n?g2..?
90: 95 d6 d0 0b c0 1d bd ff f9 ff ac 26 0a d8 bd 10  ????????.?&????
a0: 00 4b f2 00 00 00 00 00 00 00 00 00 33 00 00 c0  .K?.....3..?
b0: 00 54 00 00 00 00 60 02 00 01 ff ff 1f 71 03 00  .T....`?.?..?q?.
c0: 00 00 27 ff 00 00 00 00 00 00 00 00 00 00 00 00  ..'.....
d0: 60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ^.....
e0: 00 81 01 00 10 2b 03 1e 22 41 ff ff ff ff ff ff  .???.?+??"A.....
f0: ff 00 00 00 27 a0 00 49 d9 00 71 0f 00 80 00 80  ....'?..I?.q?..?

```

### 3. Læs et register

```
// readreg.c
// Læser et register fra i2c (BMP280)

#include <unistd.h>           //Needed for I2C port
#include <fcntl.h>           //Needed for I2C port
#include <sys/ioctl.h>       //Needed for I2C port
#include <linux/i2c-dev.h>   //Needed for I2C port
#include <stdio.h>
#include <errno.h>

// Test med fra CLI # i2cdump -y 1 0x76

void main()
{
    int file_i2c;
    int ADDR = 0x76;         //<<<<<The I2C address of the slave
    int length = 8;         //<<< Number of bytes to read
    unsigned char buffer[256] = {0};
    char *filename = (char*)"/dev/i2c-1";
    char reg[2];

    //----- OPEN THE I2C BUS -----
    if((file_i2c = open(filename, O_RDWR)) < 0)
    {
        printf("Failed to open the i2c bus\n");
        return;
    }

    if (ioctl(file_i2c, I2C_SLAVE, ADDR) < 0)
    {
        printf("Failed to acquire bus access and/or talk to slave.\n");
        return;
    }

    // Læs register 0xD0 (chipID)
    reg[0] = 0xD0;
    write(file_i2c, reg, 1); // Fortæller hvilket register vi vil læse
    if (read(file_i2c, buffer, 1) != 1)
    {
        //ERROR HANDLING: i2c transaction failed
        printf("Failed to read from the i2c bus: %d\n", errno);
    }

    printf("Chip ID = 0x%x\n", buffer[0]);
}
```

Figur 3.1 readreg.c

På min RPi er output fra readreg.c: Chip ID = 0x60

Det må betyde min ikke er en original, en original Bosh BMP280 ville svare med 0x58

