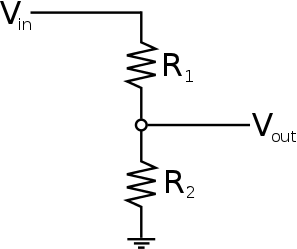


Figur 1 - bmp085\_pin\_diagram – Indsæt i afsnittet om bmp085 i arkitekturen

# Tekniske overvejelser

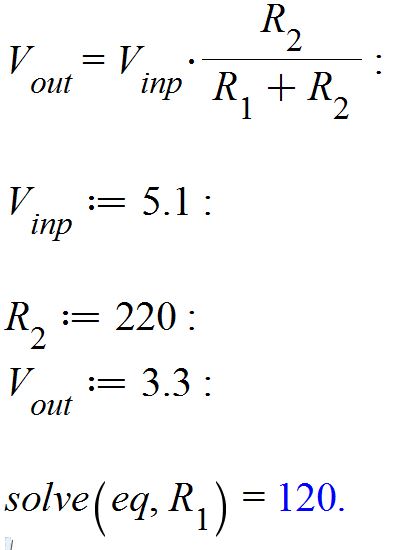
## BMP085

I forbindelse med læsning af sensordata fra den barometriske trykmåler BMP085, bruges I2C interfacet. Atmega32 understøtter I2C komunikatiuon og på figur XX ses den hardwaremæssige opsætning. Atmega 32’s input spænding er 5V bruges en simpel spændingsdeler, så BMP085 ikke overbelastes.



Figur 2 - voltage\_divider

På figur XX vise modstandsberegningerne til spændingsdeleren.



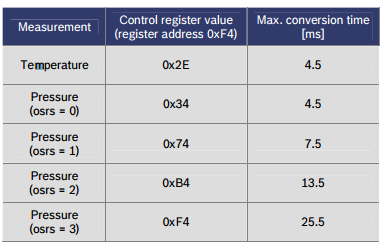
Figur 3 - voltage\_div

## Brug af i2c bussen med BMP085

I2C protokollen implementerer master/slave teknikken, og er et TWI (two wire interface).

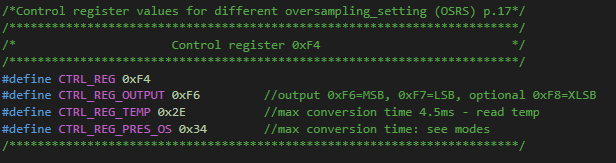
Atmega32 agerer I2C master som sætter clock hastigheden ved at skrive til PIN (SCK). I2C masteren sender data bytes til I2C slaven på addressen 0x77.

Ifølge BMP085’s datablad er følgene kommandoer mulige

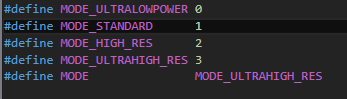


Figur 4 - bmp085\_commands

Kommandoerne er definerede BMP085 driverens h-fil. Oversampling modes er defineret separat.

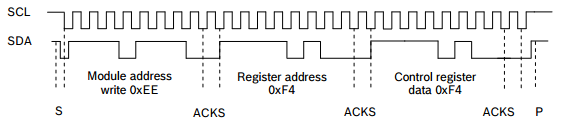


Figur 5 - bmp085\_command\_defines



Figur 6 - bmp085\_command\_modes

Timing diagrammet for start af temperatur/tyk måling fremgår a BM085’s datablad, og kan ses på figur XX.



Figur 7 - bmp\_i2c\_timing

Som det kan ses på figur 7, sender BMP085 en ACK for hver 8 bit der modtages. Atmega32 masteren sender en stop condition efter sidste ACK. S indikerer start conditionen.

Som angivet på figur 4 er der konverterings tider forbundet med målingerne. BMP085 giver muliged for at signalere et fuldendt måling ved at sætte BMP085’s EOC pin høj når en kovertering er slut. Dette bruges dog ikke i dette projekt da der ikke har været tid til at udvikle kredsløbet der hiver 3.3v logik op til 5v.

## Implementering af I2C driver

Der er i dette projekt taget udgangspunkt i en i2c driver\footnote{Peter Fleury} til at initiere interfacet samt sende/modtage tage på i2c bussen.

Af hovedfunktionaliteter der benyttes kan der blandt andet nævnes:

* Initiering af I2C bussen
  + Sætter clock hastigheden (uden prescaler)
* Afsendelse af start condition
  + TWCR = (1<<TWINT) | (1<<TWSTA) | (1<<TWEN). TWI interrupt flagget cleares, start flagget sættes, TWI Enable sættes til 1.
* Afsendelse af stop condition
  + TWCR = (1<<TWINT) | (1<<TWEN) | (1<<TWSTO);
  + Herefter frigives i2c bussen
* Indlæsning af data
  + Returnerer indholdet af TWDR. Hvis der ønskes yderligere data fra BMP085 sættes TWEA bitten (Acknowlage)

## Fejlhåndtering i I2C driveren

Når en start condition sendes til BMP085 returneres 0 hvis den ikke er tilgængelig.

## Kalibrering af BMP085

TBMP085 har indkodet kalibrerings data i dens EEPROM, som bruges til at kompensere offsettet på udførte målinger. I alt indeholder EEPROM’en 176 bit kalibreringsdata der har til formål at tweake sensorens parametre og målinger\footnot{BMP085 Data Sheet rev. 1.2 side 8}. I I2C driveren findes en funktion til indlæsning af kalibreringsdata.

For at give en ide om kalibreringens virkemåde kan et pseudo kode eksempel ses nedenfor.

Pseudokokde ekspempel på hvordan kalbreringsdata bruges til kompensering af temperatur-måledata i driveren.

//Get calibration data

readMemory(CAL\_REG\_AC5, buffer, 2);

regAC5 = buffer[0] <<8 | buffer[1]));

readMemory(CAL\_REG\_AC6, buffer, 2);

regAC6 = buffer[0] <<8 | buffer[1]));

//calculate raw temperature

Ut = uncompensatedValue

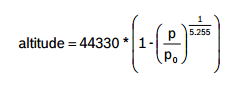
x1 = ut - regAC6) \* regAC5 >> 15;

x2 = regMC << 11) / (x1 + regMD);

rawTemperature = x1 + x2;

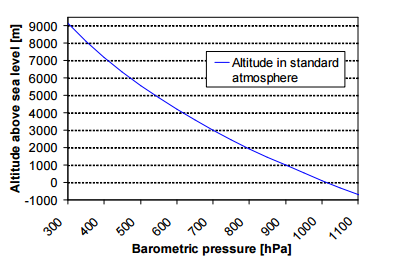
Beregningerne understøttes af BMP085’s datablad på side 13.

BMP085 har ikke mulighed for at måle altitude. Dog kan dette beregnes da det gennemsnitlige tryk ved havoverfladen er kendt da ligningen på figur \ref{eqation} er givet.



Figur 8 - equation

I driveren sættes P\_\_0 til 101325 Pa, trykket aflæses fra måleren hvorefter altitude beregnes og returneres.



Figur 9 - altitude\_graph \ref{bmp085 datasheet side 14}