

## 気温、湿度、気圧センサの データ取り込み

今回は 加速度センサでしたが、今回は 気温、湿度、気圧センサの データ取り込みを行います。マイコン環境は 前回と同じく ESP32-WROOM-32 と Arduino IDE で 行います。

今回の 気温、湿度、気圧センサ BME280はドイツの ボッシュというメーカー製品です。

ボッシュのデータシート参照値

電源電圧: 1.71 ~ 3.6V

温度測定範囲: -40 ~ +85°C

湿度測定範囲: 0 ~ 100%

気圧測定範囲: 300 ~ 1100 hPa

温度誤差: 0 ~ 65°C 範囲内で  $\pm 1^{\circ}\text{C}$

湿度誤差: 25°C 20 ~ 80% 範囲内で  $\pm 3\%$

気圧誤差: 0~65°C内にて 300 ~ 1100hPa  
範囲内で  $\pm 1.0 \text{ hPa}$

左下の 数値だけ見れば 気温、湿度、気圧の測定が出来て、精度も良いと思います。

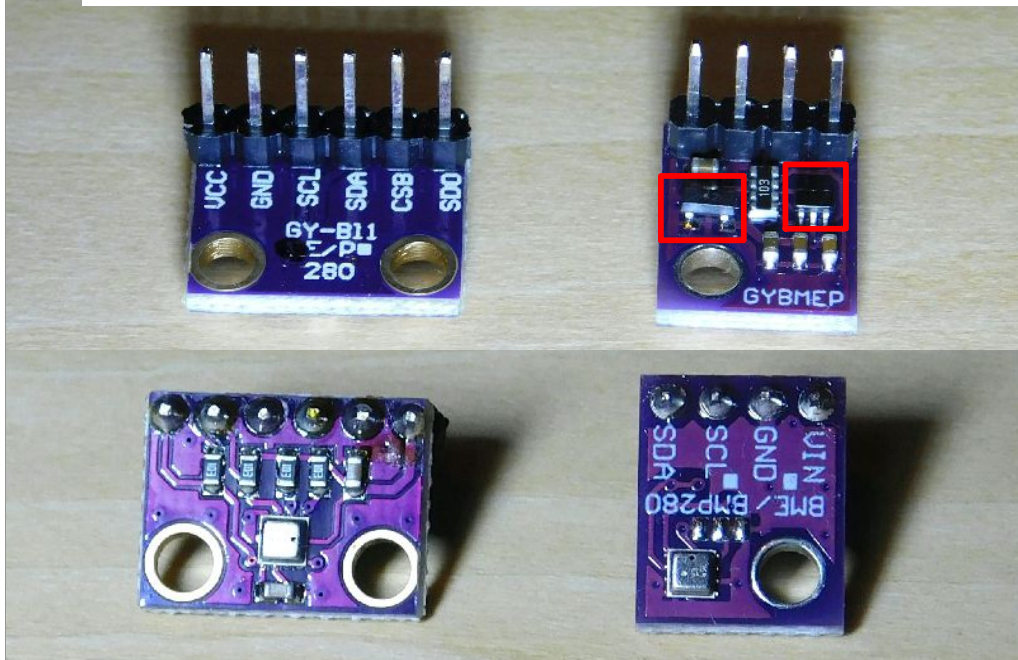
で、BME280の ICは I2Cと SPIのどちらにも対応しているようです。但し 基板が I2C 専用になっている物が 多いです。まあ、気温、湿度、気圧の測定であれば あまり高速性は必要ないので I2Cで 十分と思います。

それと注意点は 電源電圧が 1.71 ~ 3.6Vなので 3.3Vのマイコンと接続する事になります。という事で ESP32は OKです。BME280の基板は 4年ほど前に購入した記憶があるので 探したら出てきました。それも、基板形状が異なる2種類の基板が、出てきました。で、見比べると周辺回路の構成が 異なるようです。購入した amazonにて 商品サイトを調べて 2種類の基板の回路図が 見つかりました。

それらを、次のページにて お見せします。

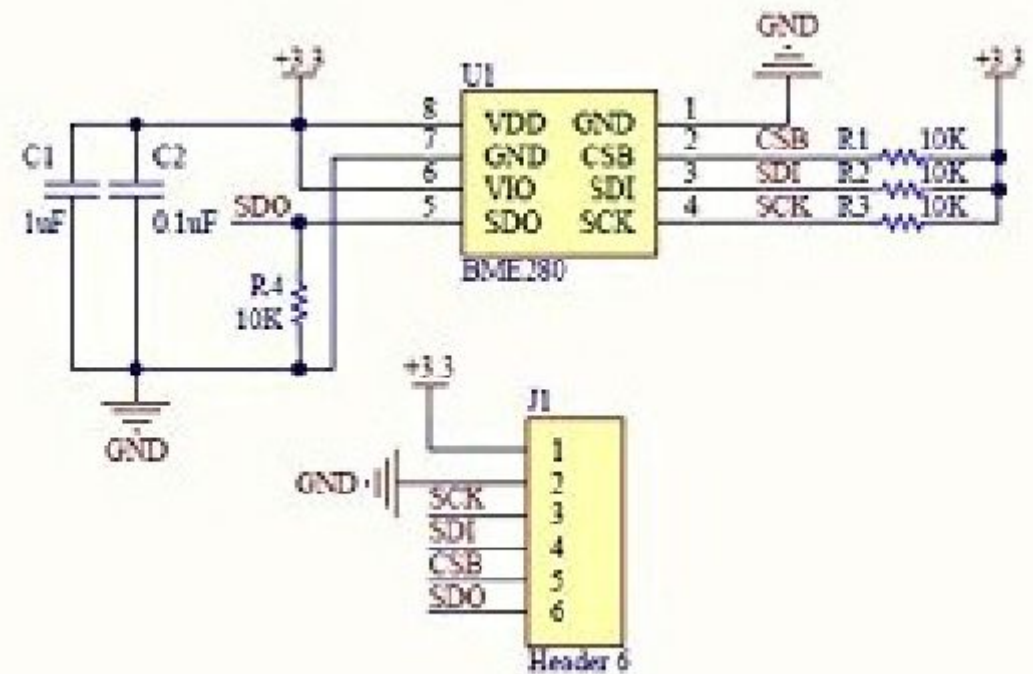
## 2種類の BME280基板

仮に、左の基板を A基板、右の基板を B基板とします。どうやらA基板は 6pinで SPIでも使えそうです。Bの基板は 4pinで I2C専用です。それとA基板は pinが見える裏側には 部品は、付いていませんが、B基板には 2つ ICらしき物が付いてます。赤枠で囲ってます。



左の赤枠は 三端子レギュレータのようです。右の赤枠は 信号線SCL、SDAのレベル変換 ICのようです。下の回路図が A基板です。

A基板には 三端子レギュレータは 無いです。



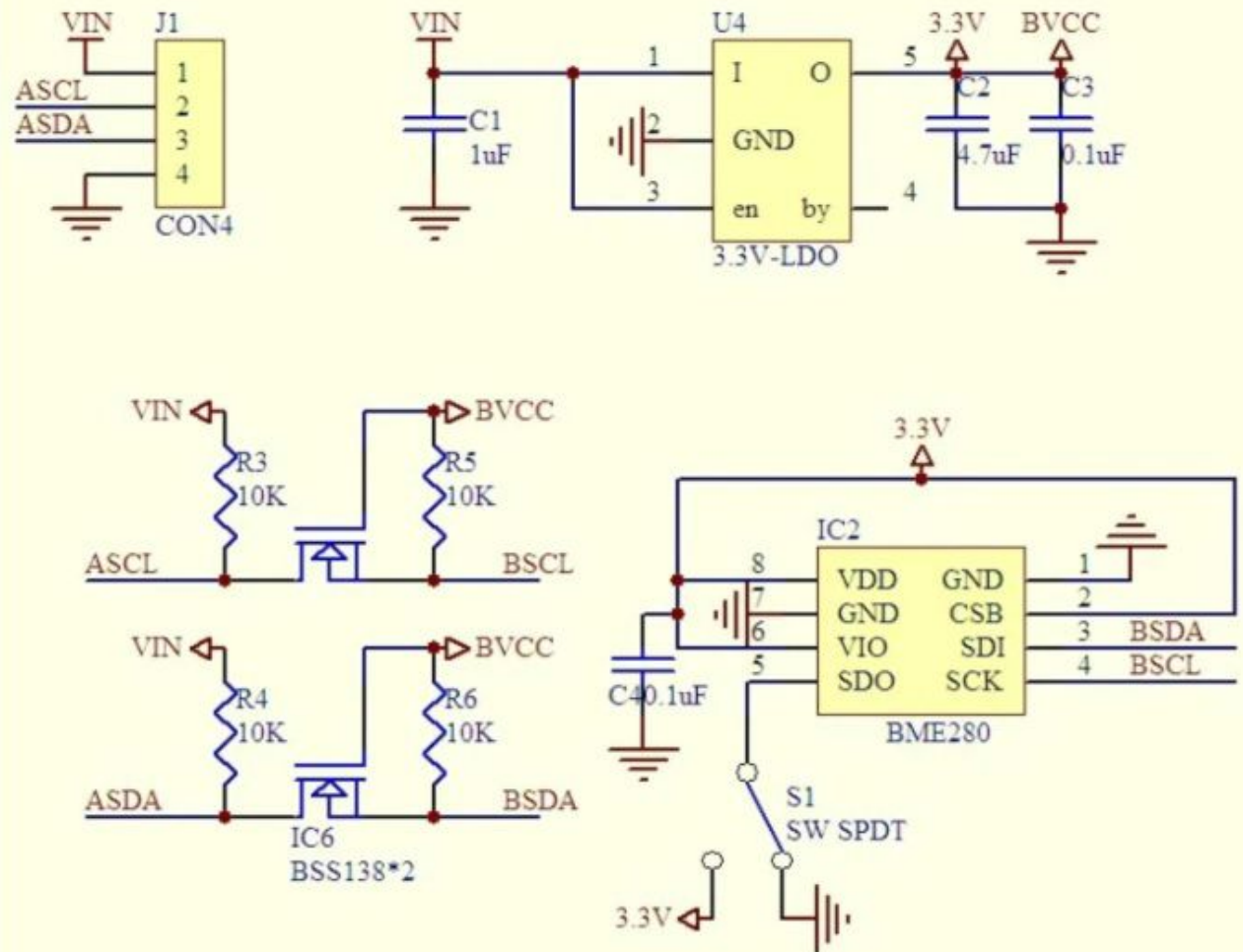
右の回路図は B基板です。  
右上の U4が Vin入力の電圧を  
3.3Vに して出力する三端子レ  
ギュレータです。 BME280に 接  
続されます。

右下 BME280の 左下側 5pin  
SDOの S1の スイッチが接続さ  
れてますが、実際の基板には あ  
りません。 SDOを 直接グランド  
に接続してあります。

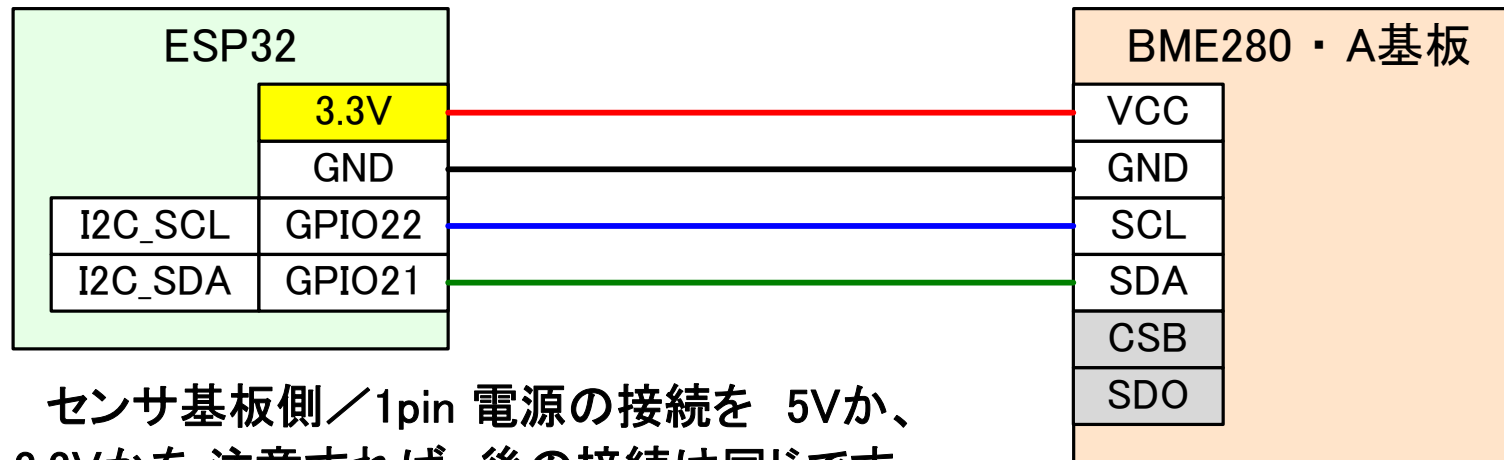
B基板 J1 1pin の Vin は 5V  
を接続する事になります。 基板  
内に BME280専用の 3.3Vのレギ  
ュレータが ある事になります。

ノイズの面では、この回路の方  
が 有利と思います。

## モジュール回路図

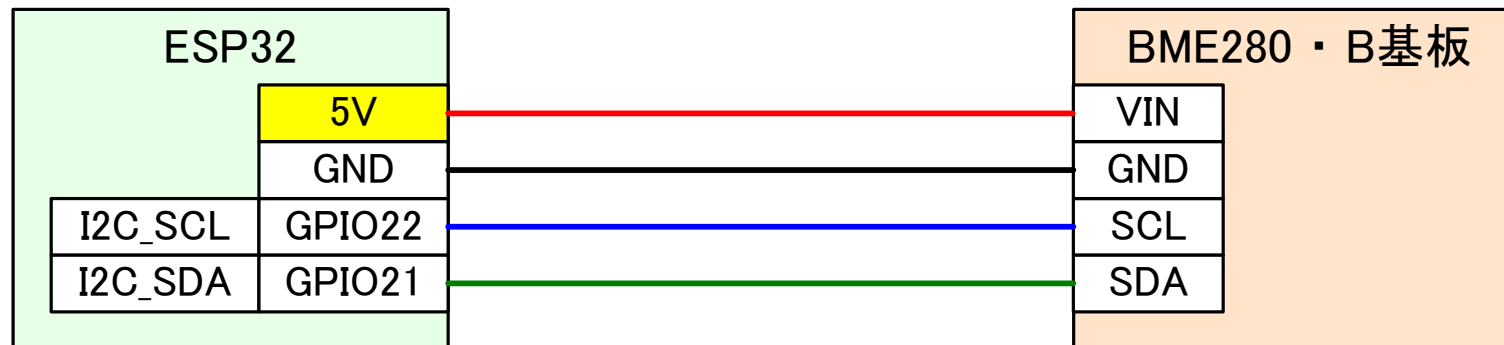


## ESP32と 2種類の BME280基板の 接続



センサ基板側／1pin 電源の接続を 5Vか、3.3Vかを 注意すれば、後の接続は同じです。

A基板の CSBと SDOは、SPI接続時の信号線です。今回は使用しません。



## BME280基板の データ読み出しに関して

BME280のデータ読み出しの プログラムに関しては、Arduino環境では BME280の アクセスを行うライブラリが 既にあり、そのライブラリを使用すると、あっさり BME280を 使用する事が、簡単に出来ます。

しかし、それってプログラムの勉強になるのかな。？ と 私は、疑問に思います。 かって いった これらを ゼロから作るのは、BME280との通信手順や 読み出したバイナリ値を 難解な補正計算を行い、温度、湿度、気圧の値に変換するのは至難の業です。 よって、中間をとって I2Cの通信機能をサポートする Wire ライブラリを使った、BME280アクセスの スケッチを使った方が BME280の I2C通信の 手順や、取り出したデータのバイナリ値の整列や 厄介な補正計算も コードが 見れます。

C言語の ソースコードが見れると 何を やっているのか、という事の理解を深める事も出来ますし、部分的に 改修する事も出来ます。 あと 別のマイコンに移植する場合も、このような ソースがあると 事がスムーズに進みます。

今回は、通信の Wireライブラリの関数を束ねて一つのまとまったアクセス関数の形に 作り直しました。 そのようなサブ関数をいくつか作成しました。 それと、前回の加速度センサのデータ読み込み直後も そうでしたが、やたら ややこしいシフト演算を行っている箇所も サブ関数にしました。 要は ビッグエンディアンと リトルエンディアンの バイト単位のデータの並べ直しです。

しかし今回は、補正計算が かなり難解で 挫折しました。



これは、BOSCHの BME280の設計に 関わった技術者が 補正計算のソースを公開したのではと 思いました。

で、BOSCHで公開している BME280に 関わる Final data sheet を 見つけました。2015年 5月7日 リリースのようです。このデータシートの 49-50ページに C言語の サンプルソースを 載せてありました。開発環境は、何か分かりませんが、32bitの環境のようです。2種類のソースを 載せてあり、49ページ側が 浮動小数点 doubleで 演算処理しており、50ページ側が、整数 int32の 整数演算で 処理しております。

Arduino UNO等で動くソースは、50ページの int32の 演算処理関数と ほぼ同じです。

何が違うかというと、32bit整数の 型宣言に BOSCHのソースでは (BME280\_S32\_T) であり、Arduinoの ソースは (int32\_t) に、なっています。計算の内容は 全く同じ様です。

という事で、今回のBME280の補正計算処理は 触らない事にします。

で、自分の分かる範囲で 改造したソースで、ビルド 書き込みして 一応動く事を確認しました。あと、BME280のセンサー基板が A基板と B基板が ありましたが、どちらも同じプログラムで 動く事を 確認しました。

という事で 動画で お見せします。