
CM-3G 周辺モジュール拡張技術文書

BME280 センサ(温度、湿度、気圧)

(第 1 版)

Copyright (C) 2015 株式会社コンピューテックス

目 次

1. はじめに.....	1
2. BME280 について.....	1
3. 接続図.....	1
4. シェル・スクリプト	2
5. シェル・スクリプトの実行.....	6

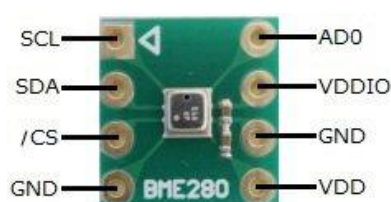
1. はじめに

本書は、CM-3G 開発キットで BME280 センサを使用するための補足マニュアルです。
CM-3G の拡張コネクタに接続可能なセンサを使用するための手順を記述します。

2. BME280 について

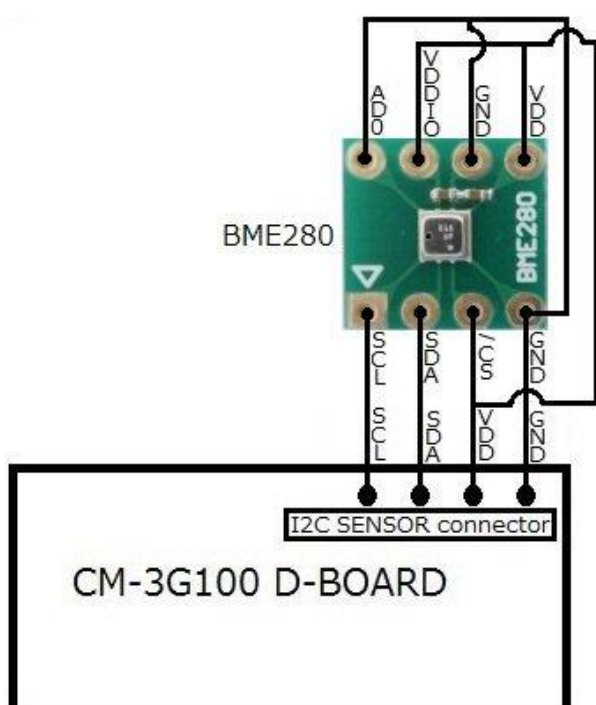
BME280 は、I2C で接続可能な温度、湿度、気圧センサです。

3. 接続図



■ BME280 と CM-3G100 D-BOARD の接続信号表

BME280	CM-3G100 D-BOARD SENSOR コネクタ
SCL	SCL
SDA	SDA
/CS	VDD
GND	GND
AD0	GND
VDDIO	VDD
VDD	VDD



4. シェル・スクリプト

センサ情報を表示するシェル・スクリプトは以下の通りです。

```
#!/bin/sh

I2CADDR=0x76

usage()
{
    echo "Usage: $0 -t -h -p"
    echo "Where,"
    echo "    -t : read Temperature"
    echo "    -h : read Humidity"
    echo "    -p : read Pressure"
    echo "Eg: $0 -t -h -p or $0 -t -h"
    exit 1
}

read_word_register()
{
    reg=$2
    val1=`i2cget -f -y 3 $1 $reg b`
    reg=$((reg + 1))
    valh=`i2cget -f -y 3 $1 $reg b`
    valh=$((valh << 8))
    echo $((valh | val1))
}

# Argument checks
if [ $# -lt 1 ]; then
    echo "Error: Parameter mismatch"
    usage $0
fi

# get the parameters
temp_read=0
humidity_read=0
pressure_read=0
while [ "$1" != "" ]; do
    case $1 in
        -t )
            temp_read=1
            ;;
        -h )
            humidity_read=1
            ;;
        -p )
            pressure_read=1
    esac
    shift
done
```

```

        ;;

        * )

        usage

    esac

    shift

done

# device check
id=`i2cget -f -y 3 $I2CADDR 0xD0 b`
if [ $? != 0 ] || [ $id != 0x60 ]; then
    echo "Error: BME280 not detected"
    exit 1;
fi

# initialize
# IIR filter (16), t-standby(20ms)
i2cset -f -y 3 $I2CADDR 0xF5 0xF0 b
# Prepare for Forced mode
i2cset -f -y 3 $I2CADDR 0xF4 0x00 b

# Temperature-compensated humidity and pressure
# Set Oversampling x 2
val=`i2cget -f -y 3 $I2CADDR 0xF4 b`
val=$((val & ~0xE0 | 0x40))
i2cset -f -y 3 $I2CADDR 0xF4 $val
# Read Compensation parameter from Non-volatile memory
dig_T1=$(read_word_register $I2CADDR 0x88)
dig_T2=$(read_word_register $I2CADDR 0x8A)
dig_T3=$(read_word_register $I2CADDR 0x8C)

if [ $humidity_read == 1 ]; then
    # Set Oversampling x 1
    i2cset -f -y 3 $I2CADDR 0xF2 0x01
    # Read Compensation parameter from Non-volatile memory
    dig_H1=`i2cget -f -y 3 $I2CADDR 0xA1 b`
    dig_H2=$(read_word_register $I2CADDR 0xE1)
    dig_H3=`i2cget -f -y 3 $I2CADDR 0xE3 b`
    dig_H4_5=`i2cget -f -y 3 $I2CADDR 0xE5 b`
    dig_H4_xmsb=`i2cget -f -y 3 $I2CADDR 0xE4 b`
    dig_H4=$((dig_H4_xmsb << 4 | (dig_H4_5 & 0x0F)))
    dig_H5_xmsb=`i2cget -f -y 3 $I2CADDR 0xE6 b`
    dig_H5=$((dig_H5_xmsb << 4 | (dig_H5_lsb >> 4)))
    dig_H6=`i2cget -f -y 3 $I2CADDR 0xE7 b`
fi

if [ $pressure_read == 1 ]; then
    # Set Oversampling x 16
    val=`i2cget -f -y 3 $I2CADDR 0xF4 b`
    val=$((val & ~0x1C | 0x14))
    i2cset -f -y 3 $I2CADDR 0xF4 $val
    # Read Compensation parameter from Non-volatile memory
    dig_P1=$(read_word_register $I2CADDR 0x8E)

```

```

dig_P2=$(read_word_register $I2CADDR 0x90)
dig_P3=$(read_word_register $I2CADDR 0x92)
dig_P4=$(read_word_register $I2CADDR 0x94)
dig_P5=$(read_word_register $I2CADDR 0x96)
dig_P6=$(read_word_register $I2CADDR 0x98)
dig_P7=$(read_word_register $I2CADDR 0x9A)
dig_P8=$(read_word_register $I2CADDR 0x9C)
dig_P9=$(read_word_register $I2CADDR 0x9E)

fi

while [ 1 ]; do
    echo "**** BME280 ****"
    # Forced mode
    val=`i2cget -f -y 3 $I2CADDR 0xF4`
    val=$(( (val & ~0x03) | 0x01 ))
    i2cset -f -y 3 $I2CADDR 0xF4 $val
    # Wait for the measurement to be completed
    while [ 1 ]; do
        val=`i2cget -f -y 3 $I2CADDR 0xF3 b`
        val=$((val & 0x08))
        if [ $val == 0 ]; then
            break;
        fi
    done
    # Temperature-compensated Pressure and humidity
    temp_msb=`i2cget -f -y 3 $I2CADDR 0xFA b`
    temp_lsb=`i2cget -f -y 3 $I2CADDR 0xFB b`
    temp_xlsb=`i2cget -f -y 3 $I2CADDR 0xFC b`
    adc_T=$(( (temp_msb << 12) | (temp_lsb << 4) | (temp_xlsb >> 4) ))
    var1=$(( (((adc_T >> 3) - (dig_T1 << 1)) * (dig_T2)) >> 11 ))
    var2=$(( ((((((adc_T >> 4) - (dig_T1)) * ((adc_T >> 4) - (dig_T1))) >> 12) * (dig_T3)) >> 14) ))
    t_fine=$((var1 + var2))
    if [ $temp_read == 1 ]; then
        temp=$(( (t_fine * 5) + 128 ) >> 8 ))
        temp=`echo $temp | awk '{printf"%0.2f°C\n", $1/100}'`
        echo "Temperature : $temp"
    fi
    if [ $humidity_read == 1 ]; then
        hum_msb=`i2cget -f -y 3 $I2CADDR 0xFD b`
        hum_lsb=`i2cget -f -y 3 $I2CADDR 0xFE b`
        adc_H=$(( (hum_msb << 8) | hum_lsb ))
        v_x1_u32r=$((t_fine - 76800))
        v_x1_u32r=$(( ((((((adc_H << 14) - ((dig_H4) << 20) - ((dig_H5) * v_x1_u32r)) + (16384))
>> 15) * ((((((v_x1_u32r * (dig_H6)) >> 10) * ((v_x1_u32r * (dig_H3)) >> 11) + (32768))) >> 10) +
(2097152)) * (dig_H2) + 8192) >> 14) )))
        v_x1_u32r=$(( (v_x1_u32r - (((v_x1_u32r >> 15) * (v_x1_u32r >> 15)) >> 7) * (dig_H1))
>> 4) ))
        v_x1_u32r=$(( (v_x1_u32r < 0 ? 0 : v_x1_u32r) ))
        v_x1_u32r=$(( (v_x1_u32r > 419430400 ? 419430400 : v_x1_u32r) ))
        v_x1_u32r=$((v_x1_u32r >> 12))
        v_x1_u32r=`echo $v_x1_u32r | awk '{printf"%0.3f %RH\n", $1/1024}'`
    fi
done

```

```

        echo "Humidity      : $v_x1_u32r"

    fi
    if [ $pressure_read == 1 ]; then
        pressure_msb=`i2cget -f -y 3 $I2CADDR 0xF7 b`
        pressure_lsb=`i2cget -f -y 3 $I2CADDR 0xF8 b`
        pressure_xlsb=`i2cget -f -y 3 $I2CADDR 0xF9 b`
        adc_P=$(( (pressure_msb << 12) | (pressure_lsb << 4) | (pressure_xlsb >> 4) ))
        var1=$(( (t_fine) >> 1) - 64000 ))
        var2=$(( (((var1 >> 2) * (var1 >> 2)) >> 11) * (dig_P6) ))
        var2=$(( (var2 + ((var1 * (dig_P5)) << 1) )) )
        var2=$(( (var2 >> 2) + ((dig_P4) << 16) ))
        var1=$(( (((dig_P3 * (((var1 >> 2) * (var1 >> 2)) >> 13)) >> 3) + (((dig_P2) * var1)
>> 1)) >> 18) ))
        var1=$(( (((((32768 + var1)) * (dig_P1)) >> 15) )) )
        if [ $var1 -eq 0 ]; then
            p="0.000 hPa"
        else
            p=$(( (((((1048576) - adc_P) - (var2 >> 12)) * 3125) )) )
            sign=$(( (p & 0x80000000) ))
            if ! [ $sign -eq 0 ]; then
                p=$(( (p << 1) / (var1) ))
            else
                p=$(( (p / var1) * 2 ))
            fi
            var1=$(( (((dig_P9) * (((p >> 3) * (p >> 3)) >> 13)) >> 12) ))
            var2=$(( (((p >> 2)) * (dig_P8)) >> 13) ))
            p=$(( (p + ((var1 + var2 + dig_P7) >> 4) )) )
            p=`echo $p | awk '{printf"%0.3f hPa\n", $1/100}'`
        fi
        echo "Pressure      : $p"
    fi
    # delay
    usleep 500000
done

```

5. シェル・スクリプトの実行

以下のようにシェル・スクリプトを実行すると、センサの値を取得して表示します。

```
# bme280.sh -t -h -p
```

シェル・スクリプトのオプションは以下の通りです。少なくとも 1 つのオプションを指定してください。

引数	内容
-t	温度表示
-h	湿度表示
-p	気圧表示

シェル・スクリプトの実行結果は以下のようになり、温度、湿度、気圧の情報を一定時間ごとに表示します。

シェル・スクリプトを終了する場合は、**[Ctrl] + [C]**を入力してください。

```
# bme280.sh -t -h -p

**** BME280 ****
Temperature : 27.70°C
Humidity    : 44.956 %RH
Pressure    : 984.180 hPa
**** BME280 ****
Temperature : 27.70°C
Humidity    : 44.801 %RH
Pressure    : 984.310 hPa
```

変更履歴

日付	版	内容
2015-08	1	初版

-
- 本書の内容の一部、または全部を無断で使用することや、複製することはできません。
 - 本書の内容、および仕様に関しては将来予告なしに変更することがあります。
 - 本書は万全の注意を払って生産されていますが、ご利用になった結果について当社は一切の責任を負いかねますのでご了承ください。
 - COMPUTEX は、(株)コンピューテックスの登録商標です
 - その他本書で取り上げる会社名および製品名などは、一般に各メーカーの商標、または登録商標です。



株式会社コンピューテックス

本 社

〒605-0846 京都市東山区五条橋東 4-432-13 対嵐坊ビル

TEL: 075-551-0528(代) FAX: 075-551-2585

東京営業所

TEL: 03-5753-9911(代) FAX: 03-5753-9917

テクニカルセンタ

TEL: 075-551-0373 FAX: 075-551-2585

CM-3G 周辺モジュール拡張技術文書
BME280 センサ(温度、湿度、気圧)
2015 年 8 月 第 1 版
CX506(A)1508
