树莓派系列教程12：I2C总线控制BMP180

2015-8-28 20:09 | 发布者: MyMX1213 | 查看: 3840 | 评论: 4 | 原作者: MyMX1213

摘要: 本章讲解如何实际通过I2C编程读取i2c接口的压强传感器BMP180

通过上一章，相信各位对树莓派I2C编程有一定的了解了，今天我们继续使用I2C来控制BMP180压强传感器。BMP180压强传感器操作原理比较简单，开机先通过I2C读取出AC1, AC2, AC3, AC4, AC5, AC6, B1, B2, MB, MC, MD等寄存器的值，这些寄存器的值作为校准时使用。如何读取温度寄存器，压强寄存器的值，根据下图公式算出测得的当前温度和压强。

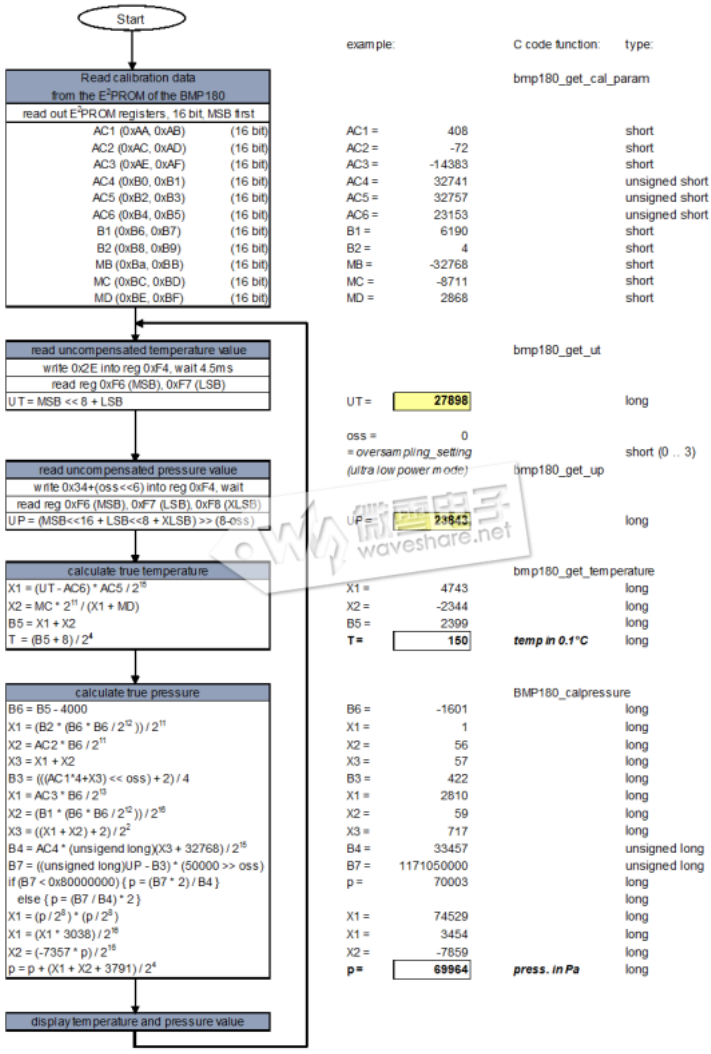

Figure 4: Algorithm for pressure and temperature measurement

本章主要讲解python程序，使大家熟悉python编程。关于bcm2835,wiringpi程序具体可参看Pioneer600示例程序。

驱动文件bmp180.py

```
001 import time
002 import smbus
003
004 # BMP085 default address.
005 BMP180_I2CADDR           = 0x77
006
007 # Operating Modes
008 BMP180_ULTRALOWPOWER     = 0
009 BMP180_STANDARD          = 1
010 BMP180_HIGHRES           = 2
011 BMP180_ULTRAHIGHRES      = 3
012
013 # BMP085 Registers
014 BMP180_CAL_AC1           = 0xAA  # R   Calibration data (16 bits)
015 BMP180_CAL_AC2           = 0xAC  # R   Calibration data (16 bits)
016 BMP180_CAL_AC3           = 0xAE  # R   Calibration data (16 bits)
```

微雪课堂
树莓派 Arduino C8051 PIC STM8 FPGA
树莓派
01 Alphabot树莓派教程
lede是openwrt的一个分
有支持pi3
02 Alphabot树莓派教程
03 树莓派系列教程18：
04 树莓派系列教程17：
05 树莓派系列教程16：
06 树莓派系列教程15：
07 树莓派系列教程14：
08 树莓派系列教程13：
09 树莓派系列教程12：
010 树莓派系列教程11
011 树莓派系列教程10
012 树莓派系列教程9：
013 树莓派系列教程8：
014 树莓派系列教程8：
015 树莓派教程系列7：
016 树莓派教程系列6：
017 树莓派系列教程5：
018 树莓派系列教程4：
019 树莓派系列教程3：
020 树莓派系列教程3：
021 树莓派系列教程3：
022 树莓派系列教程2：
023 树莓派系列教程1：

微雪课堂

```
BMP180_CAL_AC4          = 0xB0  # R   Calibration data (16 bits)
BMP180_CAL_AC5          = 0xB2  # R   Calibration data (16 bits)
BMP180_CAL_AC6          = 0xB4  # R   Calibration data (16 bits)
BMP180_CAL_B1           = 0xB6  # R   Calibration data (16 bits)
BMP180_CAL_B2           = 0xB8  # R   Calibration data (16 bits)
BMP180_CAL_MB           = 0xBA  # R   Calibration data (16 bits)
BMP180_CAL_MC           = 0xBC  # R   Calibration data (16 bits)
BMP180_CAL_MD           = 0xBE  # R   Calibration data (16 bits)
BMP180_CONTROL          = 0xF4
BMP180_TEMPDATA         = 0xF6
BMP180_PRESSUREDATA     = 0xF6

# Commands
BMP180_READTEMPCMD      = 0x2E
BMP180_READPRESSURECMD  = 0x34


class BMP180(object):
    def __init__(self, address=BMP180_I2CADDR, mode=BMP180_STANDARD):
        self._mode = mode
        self._address = address
        self._bus = smbus.SMBus(1)
        # Load calibration values.
        self._load_calibration()
    def _read_byte(self,cmd):
        return self._bus.read_byte_data(self._address,cmd)

    def _read_u16(self,cmd):
        MSB = self._bus.read_byte_data(self._address,cmd)
        LSB = self._bus.read_byte_data(self._address,cmd+1)
        return (MSB << 8) + LSB

    def _read_s16(self,cmd):
        result = self._read_u16(cmd)
        if result > 32767:result -= 65536
        return result

    def _write_byte(self,cmd,val):
        self._bus.write_byte_data(self._address,cmd,val)

    def _load_calibration(self):
        "load calibration"
        self.cal_AC1 = self._read_s16(BMP180_CAL_AC1)   # INT16
        self.cal_AC2 = self._read_s16(BMP180_CAL_AC2)   # INT16
        self.cal_AC3 = self._read_s16(BMP180_CAL_AC3)   # INT16
        self.cal_AC4 = self._read_u16(BMP180_CAL_AC4)   # UINT16
        self.cal_AC5 = self._read_u16(BMP180_CAL_AC5)   # UINT16
        self.cal_AC6 = self._read_u16(BMP180_CAL_AC6)   # UINT16
        self.cal_B1  = self._read_s16(BMP180_CAL_B1)     # INT16
        self.cal_B2  = self._read_s16(BMP180_CAL_B2)     # INT16
        self.cal_MB  = self._read_s16(BMP180_CAL_MB)     # INT16
        self.cal_MC  = self._read_s16(BMP180_CAL_MC)     # INT16
        self.cal_MD  = self._read_s16(BMP180_CAL_MD)     # INT16

    def read_raw_temp(self):
        """Reads the raw (uncompensated) temperature from the sensor."""
        self._write_byte(BMP180_CONTROL, BMP180_READTEMPCMD)
        time.sleep(0.005)  # Wait 5ms
        MSB = self._read_byte(BMP180_TEMPDATA)
        LSB = self._read_byte(BMP180_TEMPDATA+1)
        raw = (MSB << 8) + LSB
        return raw

    def read_raw_pressure(self):
        """Reads the raw (uncompensated) pressure level from the sensor."""
        self._write_byte(BMP180_CONTROL, BMP180_READPRESSURECMD + (self._mode << 6))
        if self._mode == BMP180_ULTRALOWPOWER:
            time.sleep(0.005)
        elif self._mode == BMP180_HIGHRES:
            time.sleep(0.014)
        elif self._mode == BMP180_ULTRAHIGHRES:
            time.sleep(0.026)
        else:
            time.sleep(0.008)
        MSB = self._read_byte(BMP180_PRESSUREDATA)
        LSB = self._read_byte(BMP180_PRESSUREDATA+1)
        XLSB = self._read_byte(BMP180_PRESSUREDATA+2)
        raw = ((MSB << 16) + (LSB << 8) + XLSB) >> (8 - self._mode)
        return raw

    def read_temperature(self):
        """Gets the compensated temperature in degrees celsius."""
        UT = self.read_raw_temp()

        X1 = ((UT - self.cal_AC6) * self.cal_AC5) >> 15
        X2 = (self.cal_MC << 11) / (X1 + self.cal_MD)
        B5 = X1 + X2
        temp = ((B5 + 8) >> 4) / 10.0
        return temp

    def read_pressure(self):
        """Gets the compensated pressure in Pascals."""
        UT = self.read_raw_temp()
        UP = self.read_raw_pressure()


        X1 = ((UT - self.cal_AC6) * self.cal_AC5) >> 15
        X2 = (self.cal_MC << 11) / (X1 + self.cal_MD)
        B5 = X1 + X2

        # Pressure Calculations
        B6 = B5 - 4000
        X1 = (self.cal_B2 * (B6 * B6) >> 12) >> 11
        X2 = (self.cal_AC2 * B6) >> 11
        X3 = X1 + X2
        B3 = (((self.cal_AC1 * 4 + X3) << self._mode) + 2) / 4

        X1 = (self.cal_AC3 * B6) >> 13
        X2 = (self.cal_B1 * ((B6 * B6) >> 12)) >> 16
        X3 = ((X1 + X2) + 2) >> 2
        B4 = (self.cal_AC4 * (X3 + 32768)) >> 15
```

微雪课堂

```
        B7 = (UP - B3) * (50000 >> self._mode)

        if B7 < 0x80000000:
            p = (B7 * 2) / B4
        else:
            p = (B7 / B4) * 2
        X1 = (p >> 8) * (p >> 8)
        X1 = (X1 * 3038) >> 16
        X2 = (-7357 * p) >> 16

        p = p + ((X1 + X2 + 3791) >> 4)
        return p

    def read_altitude(self, sealevel_pa=101325.0):
        """Calculates the altitude in meters."""
        # Calculation taken straight from section 3.6 of the datasheet.
        pressure = float(self.read_pressure())
        altitude = 44330.0 * (1.0 - pow(pressure / sealevel_pa, (1.0/5.255)))
        return altitude

    def read_sealevel_pressure(self, altitude_m=0.0):
        """Calculates the pressure at sealevel when given a known altitude in
        meters. Returns a value in Pascals."""
        pressure = float(self.read_pressure())
        p0 = pressure / pow(1.0 - altitude_m/44330.0, 5.255)
        return p0
```

主文件bmp180_example.py

```
#!/usr/bin/python

import time
from BMP180 import BMP180

# Initialise the BMP085 and use STANDARD mode (default value)
# bmp = BMP085(0x77, debug=True)
bmp = BMP180()

# To specify a different operating mode, uncomment one of the following:
# bmp = BMP085(0x77, 0)  # ULTRALOWPOWER Mode
# bmp = BMP085(0x77, 1)  # STANDARD Mode
# bmp = BMP085(0x77, 2)  # HIRES Mode
# bmp = BMP085(0x77, 3)  # ULTRAHIRES Mode
while True:
    temp = bmp.read_temperature()

# Read the current barometric pressure level
    pressure = bmp.read_pressure()

# To calculate altitude based on an estimated mean sea level pressure
# (1013.25 hPa) call the function as follows, but this won't be very accurate
    altitude = bmp.read_altitude()

# To specify a more accurate altitude, enter the correct mean sea level
# pressure level.  For example, if the current pressure level is 1023.50 hPa
# enter 102350 since we include two decimal places in the integer value
# altitude = bmp.readAltitude(102350)

    print "Temperature: %.2f C" % temp
    print "Pressure:    %.2f hPa" % (pressure / 100.0)
    print "Altitude:     %.2f\n" % altitude
time.sleep(1)
```


上一篇：树莓派系列教程11：I2C编程

下一篇：树莓派系列教程13：Serial串口

相关阅读

- 树莓派系列教程2：树莓派烧写镜像
- 树莓派教程系列6：文件共享(samba)
- 树莓派系列教程3：访问树莓派之SSH篇
- 树莓派教程系列7：wiringPi、bcm2835、python库安装
- 树莓派系列教程3：访问树莓派之WiFi篇

最新评论

微雪课堂

BMP180 數據不準, 如何校準?

Ezio_神仙醋 2017-1-19 15:15 引用

study

MyMX1213 2016-11-24 09:26 引用

“: 我复制了下代码，出现IO错误，不知道怎么回事。 ”

你看下有没有使能i2C，用i2cdetect -y 1 命令看下是否可以读到i2C设备地址。
可以下载我们完整的程序试下，Pioneer600 这个模块有这个程序。
http://www.waveshare.net/shop/Pioneer600.htm

游客 2016-11-21 14:34 引用

我复制了下代码，出现IO错误，不知道怎么回事。

查看全部评论(4)

验证码 换一个

评论