```
/* YourDuino.com Example Software Sketch
 16 character 2 line I2C Display
 Backpack Interface labelled "A0 A1 A2" at lower right.
 ..and
 Backpack Interface labelled "YwRobot Arduino LCM1602 IIC V1"
 MOST use address 0x27, a FEW use 0x3F
 terry@yourduino.com */

/*-----( Import needed libraries )-----*/
#include <Wire.h>  // Comes with Arduino IDE
// Get the LCD I2C Library here:
// https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads
// Move any other LCD libraries to another folder or delete them
// See Library "Docs" folder for possible commands etc.
#include <LiquidCrystal_I2C.h>

/*-----( Declare Constants )-----*/
/*-----( Declare objects )-----*/
// set the LCD address to 0x27 for a 16 chars 2 line display
// A FEW use address 0x3F
// Set the pins on the I2C chip used for LCD connections:
//                    addr, en,rw,rs,d4,d5,d6,d7,bl,blpol
LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);  // Set the LCD
I2C address

// code for bme280
#define BME280_ADDRESS 0x76
unsigned long int hum_raw,temp_raw,pres_raw;
signed long int t_fine;

uint16_t dig_T1;
 int16_t dig_T2;
 int16_t dig_T3;
uint16_t dig_P1;
 int16_t dig_P2;
 int16_t dig_P3;
 int16_t dig_P4;
 int16_t dig_P5;
 int16_t dig_P6;
 int16_t dig_P7;
 int16_t dig_P8;
 int16_t dig_P9;
 int8_t  dig_H1;
 int16_t dig_H2;
 int8_t  dig_H3;
 int16_t dig_H4;
 int16_t dig_H5;
 int8_t  dig_H6;
// ends init code for bme

/*-----( Declare Variables )-----*/
//NONE

void setup()   /*----( SETUP: RUNS ONCE )----*/
{
  Serial.begin(9600);  // Used to type in characters

  lcd.begin(16,2);   // initialize the lcd for 16 chars 2 lines, turn on
backlight

// ------- Quick 3 blinks of backlight  -------------
  for(int i = 0; i< 3; i++)
  {
    lcd.backlight();
    delay(250);
    lcd.noBacklight();
    delay(250);
  }
  lcd.backlight(); // finish with backlight on

//-------- Write characters on the display ------------------
// NOTE: Cursor Position: (CHAR, LINE) start at 0
 // lcd.setCursor(0,0); //Start at character 4 on line 0
 // lcd.print("Hello, world!");
 // delay(1000);
 // lcd.setCursor(0,1);
// lcd.print("HI!YourDuino.com");
 // delay(8000);

// Wait and then tell user they can start the Serial Monitor and type in
characters to
// Display. (Set Serial Monitor option to "No Line Ending")
  lcd.clear();
//  lcd.setCursor(0,0); //Start at character 0 on line 0
 // lcd.print("Use Serial Mon");
//  lcd.setCursor(0,1);
//  lcd.print("Type to display");

// start setup for bme280
 uint8_t osrs_t = 1;             //Temperature oversampling x 1
    uint8_t osrs_p = 1;             //Pressure oversampling x 1
    uint8_t osrs_h = 1;             //Humidity oversampling x 1
    uint8_t mode = 3;               //Normal mode
    uint8_t t_sb = 5;               //Tstandby 1000ms
    uint8_t filter = 0;             //Filter off
    uint8_t spi3w_en = 0;           //3-wire SPI Disable

    uint8_t ctrl_meas_reg = (osrs_t << 5) | (osrs_p << 2) | mode;
    uint8_t config_reg    = (t_sb << 5) | (filter << 2) | spi3w_en;
    uint8_t ctrl_hum_reg  = osrs_h;


    Wire.begin();

    writeReg(0xF2,ctrl_hum_reg);
    writeReg(0xF4,ctrl_meas_reg);
    writeReg(0xF5,config_reg);
    readTrim();
    // end setup for bme280
}/*--(end setup )---*/


void loop()   /*----( LOOP: RUNS CONSTANTLY )----*/
{
  {
    // when characters arrive over the serial port...
    if (Serial.available()) {
      // wait a bit for the entire message to arrive
      delay(100);
      // clear the screen
      lcd.clear();
      // read all the available characters
      while (Serial.available() > 0) {
        // display each character to the LCD
        lcd.write(Serial.read());
      }
    }
  }
// code for bme280
 double temp_act = 0.0, press_act = 0.0,hum_act=0.0;
    signed long int temp_cal;
    unsigned long int press_cal,hum_cal;

    readData();

    temp_cal = calibration_T(temp_raw);
    press_cal = calibration_P(pres_raw);
    hum_cal = calibration_H(hum_raw);
    temp_act = (double)temp_cal / 100.0;
    press_act = (double)press_cal / 100.0;
    hum_act = (double)hum_cal / 1024.0;
    Serial.print("TEMP : ");
    Serial.print(temp_act);
    Serial.print(" DegC  PRESS : ");
    Serial.print(press_act);
    Serial.print(" hPa  HUM : ");
    Serial.print(hum_act);
    Serial.println(" %");

    delay(1000);
    lcd.setCursor(0,0); //Start at character 0 on line 0
    lcd.print("TEMP : ");
    lcd.print(temp_act);
    lcd.print(" DegC");
     lcd.setCursor(0,1);
    lcd.print("PRESS: ");
    lcd.print(press_act);
     lcd.print(" hPa ");
    delay(1000);
    lcd.setCursor(0,0);
    lcd.print(" HUM : ");
    lcd.print(hum_act);
    lcd.print(" %");

    delay(1000);
    // end of code for bme280
}/* --(end main loop )-- */

// start functions for bme280
void readTrim()
{
    uint8_t data[32],i=0;                      // Fix 2014/04/06
    Wire.beginTransmission(BME280_ADDRESS);
    Wire.write(0x88);
    Wire.endTransmission();
    Wire.requestFrom(BME280_ADDRESS,24);       // Fix 2014/04/06
    while(Wire.available()){
        data[i] = Wire.read();
        i++;
    }

    Wire.beginTransmission(BME280_ADDRESS);    // Add 2014/04/06
    Wire.write(0xA1);                          // Add 2014/04/06
    Wire.endTransmission();                    // Add 2014/04/06
    Wire.requestFrom(BME280_ADDRESS,1);        // Add 2014/04/06
    data[i] = Wire.read();                     // Add 2014/04/06
    i++;                                       // Add 2014/04/06

    Wire.beginTransmission(BME280_ADDRESS);
    Wire.write(0xE1);
    Wire.endTransmission();
    Wire.requestFrom(BME280_ADDRESS,7);        // Fix 2014/04/06
    while(Wire.available()){
        data[i] = Wire.read();
        i++;
    }
    dig_T1 = (data[1] << 8) | data[0];
    dig_T2 = (data[3] << 8) | data[2];
    dig_T3 = (data[5] << 8) | data[4];
    dig_P1 = (data[7] << 8) | data[6];
    dig_P2 = (data[9] << 8) | data[8];
    dig_P3 = (data[11]<< 8) | data[10];
    dig_P4 = (data[13]<< 8) | data[12];
    dig_P5 = (data[15]<< 8) | data[14];
    dig_P6 = (data[17]<< 8) | data[16];
    dig_P7 = (data[19]<< 8) | data[18];
    dig_P8 = (data[21]<< 8) | data[20];
    dig_P9 = (data[23]<< 8) | data[22];
    dig_H1 = data[24];
    dig_H2 = (data[26]<< 8) | data[25];
    dig_H3 = data[27];
    dig_H4 = (data[28]<< 4) | (0x0F & data[29]);
    dig_H5 = (data[30] << 4) | ((data[29] >> 4) & 0x0F); // Fix 2014/04/06
    dig_H6 = data[31];                                   // Fix 2014/04/06
}
void writeReg(uint8_t reg_address, uint8_t data)
{
    Wire.beginTransmission(BME280_ADDRESS);
    Wire.write(reg_address);
    Wire.write(data);
    Wire.endTransmission();
}


void readData()
{
    int i = 0;
    uint32_t data[8];
    Wire.beginTransmission(BME280_ADDRESS);
    Wire.write(0xF7);
    Wire.endTransmission();
    Wire.requestFrom(BME280_ADDRESS,8);
    while(Wire.available()){
        data[i] = Wire.read();
        i++;
    }
    pres_raw = (data[0] << 12) | (data[1] << 4) | (data[2] >> 4);
    temp_raw = (data[3] << 12) | (data[4] << 4) | (data[5] >> 4);
    hum_raw  = (data[6] << 8) | data[7];
}


signed long int calibration_T(signed long int adc_T)
{

    signed long int var1, var2, T;
    var1 = ((((adc_T >> 3) - ((signed long int)dig_T1<<1))) * ((signed long
int)dig_T2)) >> 11;
    var2 = (((((adc_T >> 4) - ((signed long int)dig_T1)) * ((adc_T>>4) -
((signed long int)dig_T1))) >> 12) * ((signed long int)dig_T3)) >> 14;

    t_fine = var1 + var2;
    T = (t_fine * 5 + 128) >> 8;
    return T;
}
```

```
unsigned long int calibration_P(signed long int adc_P)
{
    signed long int var1, var2;
    unsigned long int P;
    var1 = (((signed long int)t_fine)>>1) - (signed long int)64000;
    var2 = (((var1>>2) * (var1>>2)) >> 11) * ((signed long int)dig_P6);
    var2 = var2 + ((var1*((signed long int)dig_P5))<<1);
    var2 = (var2>>2)+(((signed long int)dig_P4)<<16);
    var1 = (((dig_P3 * (((var1>>2)*(var1>>2)) >> 13)) >>3) + ((((signed long
int)dig_P2) * var1)>>1))>>18;
    var1 = ((((32768+var1))*((signed long int)dig_P1))>>15);
    if (var1 == 0)
    {
        return 0;
    }
    P = (((unsigned long int)(((signed long int)1048576)-adc_P)-
(var2>>12)))*3125;
    if(P<0x80000000)
    {
       P = (P << 1) / ((unsigned long int) var1);
    }
    else
    {
        P = (P / (unsigned long int)var1) * 2;
    }
    var1 = (((signed long int)dig_P9) * ((signed long int)(((P>>3) *
(P>>3))>>13)))>>12;
    var2 = (((signed long int)(P>>2)) * ((signed long int)dig_P8))>>13;
    P = (unsigned long int)((signed long int)P + ((var1 + var2 + dig_P7) >> 4));
    return P;
}

unsigned long int calibration_H(signed long int adc_H)
{
    signed long int v_x1;

    v_x1 = (t_fine - ((signed long int)76800));
    v_x1 = (((((adc_H << 14) -(((signed long int)dig_H4) << 20) - (((signed long
int)dig_H5) * v_x1)) +
              ((signed long int)16384)) >> 15) * (((((((v_x1 * ((signed long
int)dig_H6)) >> 10) *
              (((v_x1 * ((signed long int)dig_H3)) >> 11) + ((signed long int)
32768))) >> 10) + (( signed long int)2097152)) *
              ((signed long int) dig_H2) + 8192) >> 14));
   v_x1 = (v_x1 - (((((v_x1 >> 15) * (v_x1 >> 15)) >> 7) * ((signed long
int)dig_H1)) >> 4));
   v_x1 = (v_x1 < 0 ? 0 : v_x1);
   v_x1 = (v_x1 > 419430400 ? 419430400 : v_x1);
   return (unsigned long int)(v_x1 >> 12);
}

/* ( THE END ) */
```