

Płytki interfejsu dla czujnika wilgotności SHT31

SHT31 jest zintegrowanym czujnikiem przeznaczonym do pomiaru wilgotności względnej i temperatury. Wyróżnia się doskonałymi parametrami i miniaturową obudową. Zaprojektowana płytki interfejsu pozwala wygodnie podłączyć czujnik np. do płytek Arduino lub Nucleo. Proste procedury odczytu danych ułatwią zastosowanie czujnika we własnych konstrukcjach.

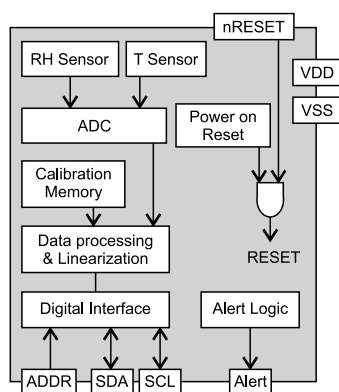
Czujnik SHT31 mierzy wilgotność względną, ale co oznacza ta wielkość? W skrócie jest to procentowy stosunek ilości wody (pary) w powietrzu do jej maksymalnej ilości (stanu nasycenia równego 100%) jaki może wystąpić w danych warunkach. Wilgotność względna zależy od temperatury. Wraz z jej spadkiem dla takiej samej ilości pary, wilgotność względna będzie się zwiększać, a po osiągnięciu 100% para zacznie się skraplać.

Informacja o poziomie wilgotności powietrza jest ważna ze względów zdrowotnych. Okazuje się, że najkorzystniej jest przebywać w środowisku, którego wilgotność względna (%RH) mieści się w przedziale 40%RH...60%RH. Poziom wilgotności jest także ważny przy długotrwałym przechowywaniu żywności, materiałów, dla niektórych procesów chemicznych.

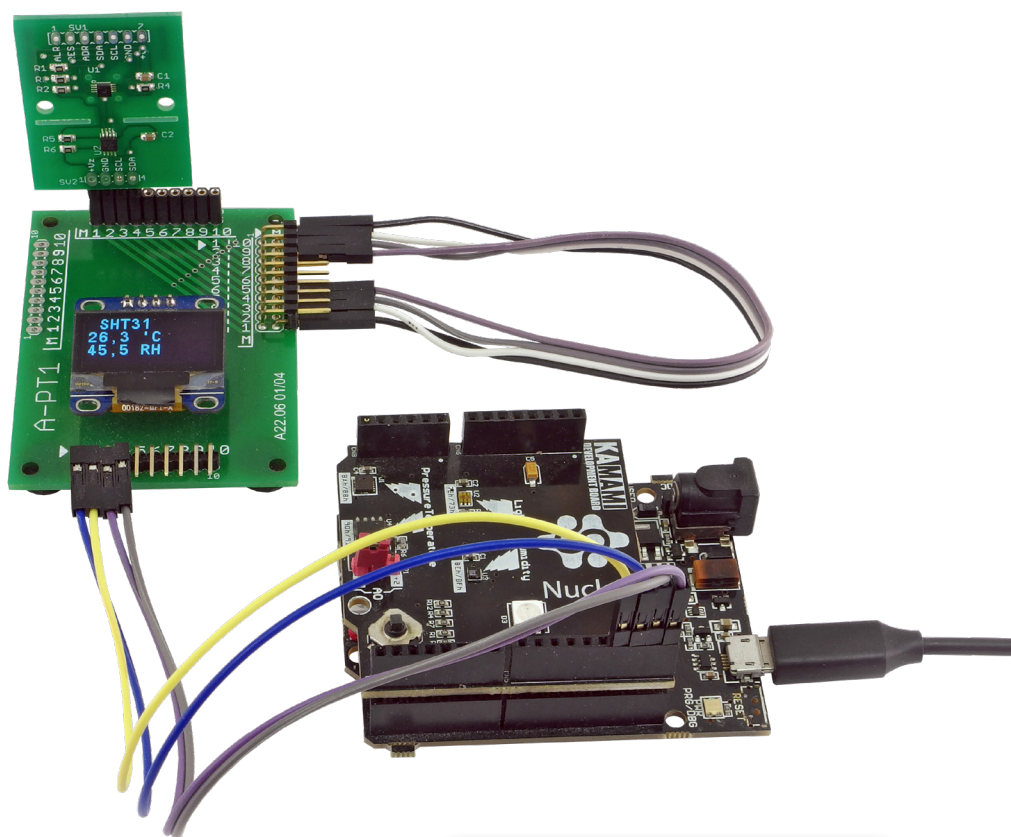
Budowa SHT31

Na rysunku 1 została pokazana budowa wewnętrzna czujnika SHT31 oraz dostępne wyprowadzenia. Oprócz podłączeń zasilania VDD, VSS, linii zerowania nRESET i wyprowadzeń magistrali I²C SDA, SCL, dostępne są jeszcze 2 dodatkowe linie.

- ADDR – wejście wyboru jednego z dwu adresów I²C, na który będzie reagował czujnik. Pozwala to na obsługę większej liczby czujników SHT31 dołączonych do tej samej magistrali;



Rysunek 1. Struktura wewnętrzna SHT31



- ALERT – wyprowadzenie służy do sygnalizacji przekroczenia zaprogramowanych progów wilgotności i temperatury. Poziomem aktywnym jest stan wysoki.

Płytki interfejsu

Niewielkie rozmiary obudowy przystosowanej do montażu powierzchniowego utrudniają bezpośrednie wykorzystanie SHT31. Specjalnie zaprojektowana płytki interfejsu pozwala względnie łatwo przyłutować czujnik, a łączya o rozdzielności styków 2,54 mm zapewniają wygodny dostęp do jego wyprowadzeń. Schemat ideowy interfejsu pokazano na rysunku 2, natomiast schemat płytki PCB pokazuje rysunek 3.

Na płytce, oprócz czujnika U1, jest miejsce na dodatkowe oporniki: R1, R2 podciągają do napięcia zasilania magistralę I²C, R3 ustawia poziom niski na linii wyboru adresu ADDR, a R4 zabezpiecza linię nRESET przed przypadkowym zerowaniem. Wszystkie wyprowadzenia sygnałowe i zasilające można podłączyć poprzez złącze SV1 bezpośrednio do systemu mikroprocesorowego. Ze względu na specyfikę magistrali I²C takie podłączenie nie może być dłuższe niż kilkanaście centymetrów. Jeżeli czujnik wilgotności ma pracować w pewnym oddaleniu od procesora sterującego, należy zastosować układ pozwalający wydłużyć magistralę.

Na płytce interfejsu przewidziano miejsce dla mostka I²C oznaczonego na schemacie

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5832

Podstawowe parametry:

- zakres pomiaru wilgotności: 0...100% RH;
- dokładność pomiaru wilgotności: ±2% RH;
- zakres pomiaru temperatury: -40...+120°C;
- dokładność pomiaru temperatury: ±0,2°C w zakresie 0...90°C;
- czas wykonywania pomiaru: od 2,5 ms do 15,5 ms;
- pobór prądu podczas pomiaru: 1500 µA;
- pobór prądu pomiędzy pomiarami: 6 µA;
- napięcie zasilania: 2,15...5,5 V;
- interfejs komunikacyjny: I²C;
- wymiary obudowy: 2,5×2,5×0,9 mm.

Wykaz elementów:

R1..R4: 10 kΩ SMD0805
 R5, R6: 10 kΩ SMD0805*
 C1 100 nF 0805
 C2 100 nF 0805*
 U1: SHT31-DIS-P
 U2: PCA9507 (TSSOP8)*
 SV1, SV4: goldpiny proste raster 2,54 mm
 * elementy montowane opcjonalnie

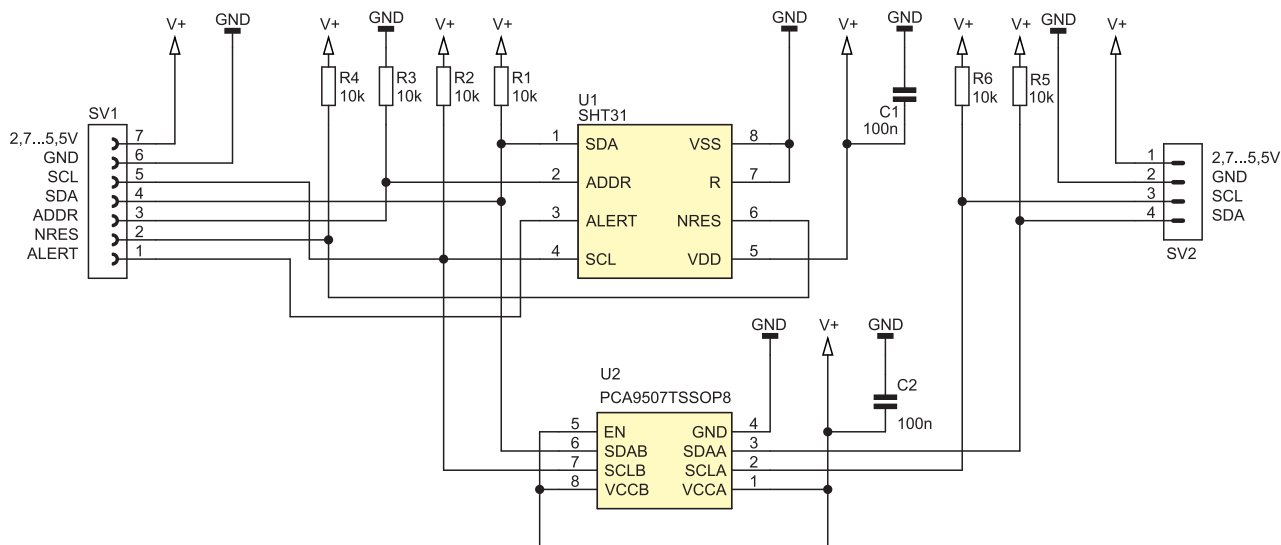
Uwaga! Elektroniczne zestawy do samodzielnego montażu. Wymagana umiejętność lutownia!

Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wylutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu.

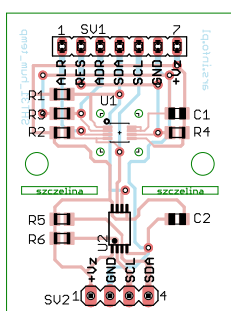
Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wylutowane w płytkę PCB)
- wersja [A] – płytki drukowana bez elementów i dokumentacji kity w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- wersja [A+] – płytki drukowana [A] + zaprogramowany układ [UK] i dokumentacja
- wersja [UK] – zaprogramowany układ

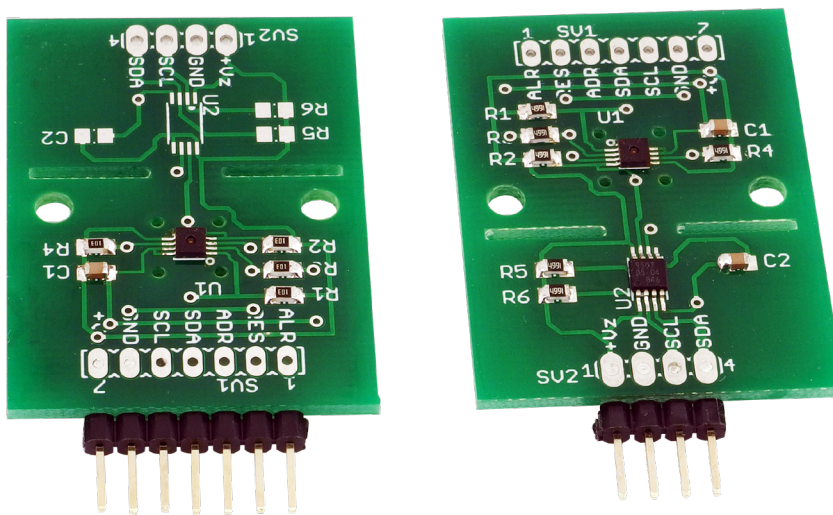
Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! <http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.



Rysunek 2. Schemat ideowy



Rysunek 3. Schemat płytki PCB



Fotografia 1. Zmontowane płytki w obydwu wariantach

symbolem U2. Zastosowanie jako mostka układu PCA9507, podniesienie napięcia zasilania do +5 V i zastosowanie po drugiej stronie magistrali od strony mikroprocesora podobnego mostka, pozwoli zwiększyć jej długość nawet do 18 metrów. Wybierając taki sposób komunikacji z płytką interfejsu, należy skorzystać ze złącza SV2. W tym wariantcie połączenia dostęp do pozostałych sygnałów sterujących czujnikiem nie jest możliwy. Jeżeli opcja z mostkiem I²C nie będzie wykorzystywana, można nie montować na płycie interfejsu elementów U2, C2, R5, R6, SV2. Na fotografii 1 pokazano płytki zmontowane w obydwu wariantach.

Cztery otwory w pobliżu miejsca wlotowania czujnika przeznaczone są do założenia osłony Filter Cap SF2. Osłona z cienką półprzepuszczalną membraną zapewnia dodatkową ochronę czujnika przed działaniem wody i substancji chemicznych, niekorzystnie wpływających na jego działanie.

Do przyłutowania czujnika potrzebna jest lutownica o cienkim grocie, topnik w płynie i odrobina cyny. Czujniki w wykonaniu SHT-31-DIS-P zabezpieczone są pomarańczową folią. Należy ją usunąć dopiero po zakończeniu montażu i oczyszczeniu płytki.

wyberane poziomem na wyprowadzeniu ADDR. Poziom niski (ustawiony w module przez opornik R3) wybiera adres 0x44, poziom wysoki wybiera adres 0x45. Adresy podane są w konwencji 7-bitowej.

Ogólny schemat transmisji rozkazu sterującego (np. inicjacji pomiaru) jest następujący: po sekwencji Start wysłany jest adres czujnika z dodanym wyzerowanym 8-bitem, a potem wysyłane są 2 bajty komendy ze starszym bajtem jako pierwszym. Transmisję kończy sekwencja Stop. Na **rysunku 4** pokazany jest przebieg transmisji rozkazu.

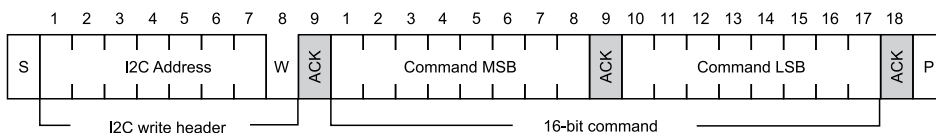
Ogólny schemat odczytu z czujnika jest następujący: po sekwencji Start wysłany jest adres czujnika z dodanym ustawionym 8-bitem, a potem czujnik przesyła 2 bajty danych temperatury (starszy bajt jako pierwszy), 1 bajt sumy kontrolnej bajtów

temperatury, 2 bajty danych wilgotności (starszy bajt jako pierwszy), 1 bajt sumy kontrolnej bajtów wilgotności. Transmisję kończy sekwencja Stop. Na **rysunku 5** pokazany jest przebieg odczytu.

Czujnik SHT31 może pracować w różnych trybach, co wpływa na sposób odczytu pomiarów. W trybie single shot realizowany jest pomiar jednorazowy, w przeciwieństwie do periodic mode, kiedy po zakończeniu poprzedniego pomiaru czujnik automatycznie rozpocznie kolejny. W obu trybach można ustawić jedną z trzech szybkości konwersji: high, medium, low, przy czym wolniejsze konwersje zapewniają dokładniejsze, bardziej powtarzalne wyniki pomiaru. W trybie automatycznego powtarzania pomiarów można ustawić liczbę pomiarów na sekundę: 0,5, 1, 2, 4, 10.

Komunikacja z czujnikiem

Komunikacja i sterowanie czujnikiem odbywają się za pośrednictwem magistrali I²C. SHT31 reaguje na dwa adresy



Rysunek 4. Transmisja rozkazu magistralą I²C

Tryb pomiaru		HEX	
Powtarzalność/Szybkość	Clock stretching	MSB	LSB
High	enabled	0x2C	06
Medium			0D
Low			10
High	disabled	0x24	00
Medium			0B
Low			16

Odczyt wyniku pomiaru zawsze inicjowany jest przez użytkownika. W trybie pojedynczego pomiaru czujnik będzie sygnalizował brak gotowości danych przez wystawienie sygnału NACK (braku potwierdzenia) po odbiorze sekwencji Start i adresu. W trybie clock stretching czujnik zawsze wysyła potwierdzenie ACK, ale gdy dane są niegotowe, blokuje sygnały

zegarowe na linii CLK magistrali I²C. Przykładowo w tabeli 1 zebrano rozkazy pojedynczego odczytu z wyborem parametrów powtarzalności i clock stretching. Poza tym czujnik reaguje na kilka dodatkowych rozkazów wpływających na sposób jego funkcjonowania np.:

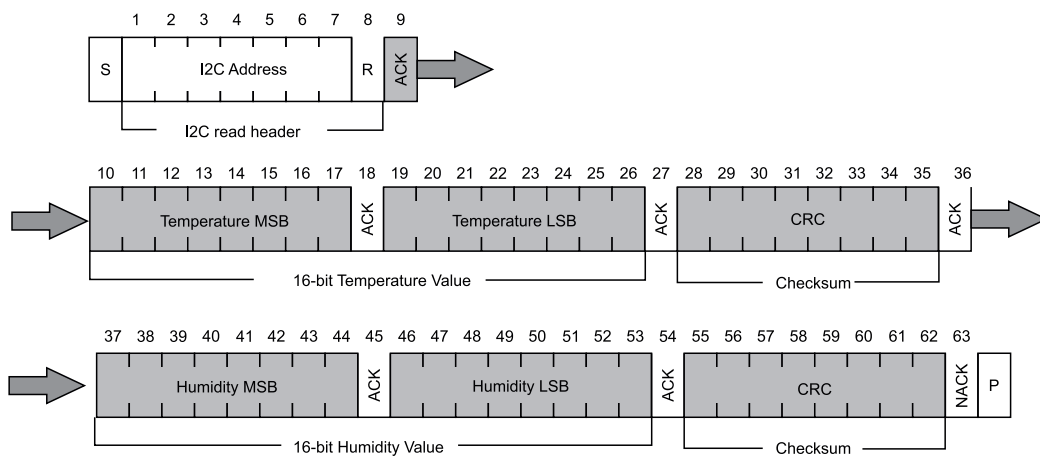
- kod 0x3093 – przerwanie trybu periodic mode,

Procedury prostego miernika wilgotności

Łącząc płytkę interfejsu z czujnikiem SHT31 oraz płytkę procesora (na zdjęciu tytułowym NUCLEO-F411), można szybko zbudować prosty miernik wilgotności połączony z cyfrowym termometrem. Linie SDA i SCL magistrali I²C płytki czujnika należy połączyć z wyprowadzeniami na płytce NUCLEO: D15

-linia SCL i D14 -linia SDA interfejsu I²C1. Do tych samych wyprowadzeń podłączamy linie I²C wyświetlacza graficznego OLED 128×64 ze sterownikiem SSD1306, na którym będą wyświetlane odczyty pomiarów. Napięcie zasilania +3,3 V dla płytki czujnika i wyświetlacza pobieramy z płytki NUCLEO.

W pętli głównej co sekundę inicjowany jest pojedynczy pomiar temperatury i wilgotności. Wywoływana jest procedura OdczytTemperaturyIWilgotności_Single, pokazana na listingu 1. Najpierw



Rysunek 5. Odczyt danych magistralą I²C

```

Listing 1.
#define SHT3I2CADDR_L7B 0x44 //I2C adres 7 bitów ADDR (pin 2) connected to logic low
//odczyt rejestrów temperatury i wilgotności
uint8_t OdczytTemperaturyIWilgotności_Single(uint8_t *p_bufor, double *temperatura, double *wilgotnosc){
#define REPEAT_MED_CLOCKSTRETCH_DIS 0x240B
uint8_t sukces;
unsigned char slave_adr, ile_bajtow_danych;
char bufor[0];

slave_adr =SHT3I2CADDR_L7B<<1;
ile_bajtow_danych =2;
bufor[0] =REPEAT_MED_CLOCKSTRETCH_DIS>>8;
bufor[1] =0xff & REPEAT_MED_CLOCKSTRETCH_DIS;

sukces =I2C_ZapisBloku(slave_adr, MEM_ADRES_I2C_0BAJT, 0, ile_bajtow_danych, bufor);
if (sukces ==FALSE) return FALSE;

ile_bajtow_danych =6;
HAL_Delay(20); //czas konwersji

sukces =I2C_OdczytBloku(slave_adr, MEM_ADRES_I2C_0BAJT,
0, ile_bajtow_danych, bufor);
if (sukces ==FALSE) return FALSE;

char crc8 =CRC8((char *)bufor,2);
char crc8_odczyt =*(bufor+2);
if (crc8 !=crc8_odczyt) return FALSE;
*wilgotnosc =KonwersjaSurowychDanychnaWilgotnosc((uint8_t *)bufor+3));

*p_bufor =bufor[0];
*(p_bufor+1) =bufor[1];
*(p_bufor+2) =bufor[3];
*(p_bufor+3) =bufor[4];

*temperatura =KonwersjaSurowychDanychnaTemperature((uint8_t *)bufor);

return TRUE;
}
    
```

adres czujnika, który będzie wywoływany, jest zmieniany z postaci 7-bitowej na 8-bitową, ponieważ zastosowana procedura zapisu i odczytu I²C wymaga takiego formatu tego parametru. Wysyłany jest rozkaz inicjujący pojedynczą konwersję. Kod rozkazu 0x240B jest opisany w tabeli 1. Procedura nie korzysta z metod odpytywania czujnika o dostępność wyniku. Po prostu odczekuje maksymalny okres potrzebny na przeprowadzenie konwersji, czyli 20 ms. Po tym czasie odczytuje z SHT31 6 bajtów danych: 2 bajty danych temperatury z bajtem sumy kontrolnej i 2 bajty danych wilgotności z bajtem sumy kontrolnej. Następuje sprawdzenie, czy obliczone sumy kontrolne danych temperatury i wilgotności są identyczne z przesłanymi, procedura obliczania CRC jest zamieszczona na **listingu 2**.

Teraz należy przekształcić odczytane surowe dane temperatury i wilgotności na stopnie °C i procenty %RH. Dla danych temperatury korzysta się z formuły:

$$T[°C] = -45 + (175 * (S_{\text{dane temperatury}} / (2^{16} - 1)))$$

Podprogram, który realizuje to przekształcenie, jest pokazany na **listingu 3**.

Dla danych wilgotności korzysta się z formuły:

$$RH = 100 * (S_{\text{dane wilgotności}} / (2^{16} - 1))$$

Podprogram, który realizuje to przekształcenie, jest pokazany na **listingu 4**.

W przypadku sukcesu główna procedura zwraca wartość TRUE i w pętli głównej otrzymane po przekształceniach wartości temperatury i wilgotności zostają wyświetlane na wyświetlaczu.

Dodatkowe informacje na temat czujnika SHT31 można znaleźć na stronie producenta: <http://bit.ly/35rNjEf>.

Ryszard Szymaniak
biuro@ars.info.pl

Listing 2.

```
char CRC8(const char *data, int length){
    uint8_t crc = 0xff;
    size_t i, j;
    for (i = 0; i < length; i++) {
        crc ^= data[i];
        for (j = 0; j < 8; j++) {
            if ((crc & 0x80) != 0)
                crc = (uint8_t)((crc << 1) ^ 0x31);
            else
                crc <<= 1;
        }
    }
    return crc;
}
```

Listing 3.

```
//konwersja wartości odczytanych rejestrów na temperaturę
double KonwersjaSurowychDanychnaTemperature(uint8_t *p_buf){
    uint16_t reg_raw = *p_buf << 8;
    reg_raw = reg_raw + *(p_buf+1);
    double temperatura = reg_raw;
    temperatura = temperatura / (65536-1);
    temperatura = temperatura * 175;
    temperatura = temperatura + (-45);
    return temperatura;
}
```

Listing 4.

```
//konwersja wartości odczytanych rejestrów na wilgotność
double KonwersjaSurowychDanychnaWilgotnosc(uint8_t *p_buf){
    uint16_t reg_raw = *p_buf << 8;
    reg_raw = reg_raw + *(p_buf+1);
    double wilgotnosc = reg_raw;
    wilgotnosc = wilgotnosc / (65536-1);
    wilgotnosc = wilgotnosc * 100;
    return wilgotnosc;
}
```