

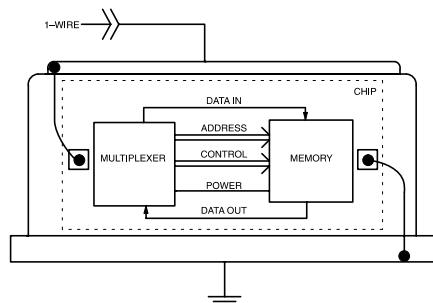
Rubryka ta powstała z myślą o rozwiązywaniu problemów, jakie najczęściej spotykają programiści. W tym „kąciku” będziemy się starać w miarę przystępnie przedstawiać rozwiązania tych problemów. Zatem zachęcamy wszystkich Czytelników do ich zgłaszania.

Bascom i 1-Wire



Gdy magistrala 1Wire jest wolna (czyli nie odbywa się żadna transmisja), na linię danych jest podawane napięcie +5 V, co powoduje naładowanie wewnętrznego kondensatora buforującego zasilanie w układzie dołączonym do magistrali (uproszczony schemat blokowy ilustrujący budowę układu DS1990 pokazano na rys. 1). Ładunek zgromadzony w kondensatorze jest wykorzystywany do zasilania układu podczas transmisji. Niektóre układy z interfejsem 1Wire mają dodatkowe wyprowadzenie, do którego jest podłączane dodatkowe napięcie zasilające +5 V. Jest to rozwiązanie stosowane przy transmisjach na duże odległości, które znacząco skraca czas przesyłania informacji (np. w DS18B20 z 750 ms do 250 ms).

Przy projektowaniu urządzeń z użyciem 1Wire trzeba wziąć pod uwagę kwestie zasilania oraz czasu odczytu i konwersji parametrów. Najczęściej stosuje się metodę zasilania z tzw. rezystorem pull-up. Do podłączenia układów 1Wire na krótkie odległości (20...30 m) możemy zastosować zwykły przewód telefoniczny, natomiast dla połączeń dłuższych zdecydowanie korzystniej jest użyć skrętki (Dallas zaleca UTP kategorii 5, lecz



Rys. 1

W tym odcinku zajmiemy się opisem dostępu do układów dołączonych do magistrali 1Wire z poziomu Bascoma.

Jak pewnie część z Czytelników EP wie, jest to magistrala zaprojektowana przez firmę Dallas (obecnie Maxim-Dallas).

Magistrala ta używa jednego przewodu do zasilania i przesyłania danych oraz oczywiście przewodu masy.



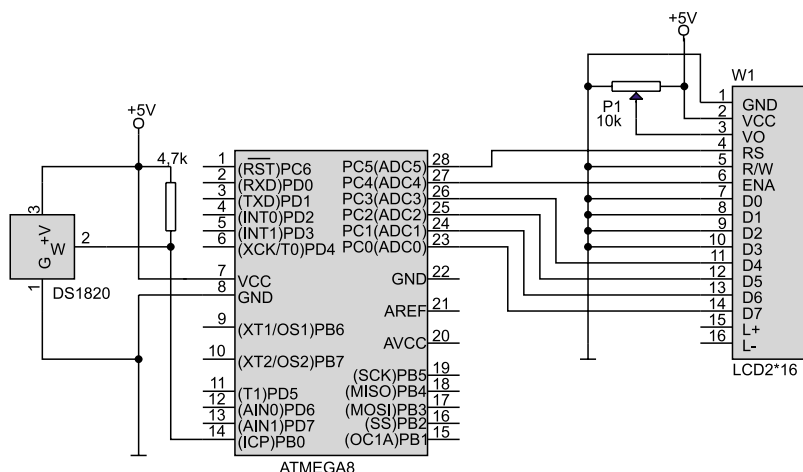
Fot. 2

zwykła skrętka, np. alarmowa, także będzie poprawnie działać). Według informacji udostępnionych przez autora BASCOM-a procedura przeszukiwania magistrali umożliwiła znalezienie do około 50 elementów na sekundę, a odczytywanie danych z różnych układów powinno być możliwe dla 13 urządzeń na sekundę. W nocie katalogowej DS1990A można znaleźć informację, że odczyt numeru serijnego pastylki nie powinien przekroczyć 10 ms.

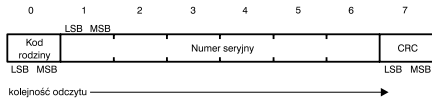
Do magistrali 1Wire można podłączyć wiele układów, takich jak np. czujniki wartości nieelektrycznych (np. opisany w dalszej części artykułu czujnik temperatury DS1820), potencjometry, przełączniki, a także układy iButton (czyli układy montowane w metalowych kapsułkach, jak na fot. 2).

Do zilustrowania sposobu obsługi magistrali 1Wire wykorzystamy ciesząc się od lat popularnością termometr DS1820 oraz jego zmodyfikowaną, unowocześnioną wersję – DS18B20. Układ DS1820 jest przedstawicielem scalonych termometrów z serii 1Wire, podłączanych do mikrokontrolera za pomocą jednej linii I/O. Typowy sposób podłączenia układu DS1820 do procesora przedstawiono na rys. 3.

Jak widać, do pracy układu jest jeszcze potrzebny rezystor „podciągający” linię danych I/O do plusa zasilania. Nie jest on co prawda niezbędny, jednak producent zaleca takie właśnie podłączenie. Z doświadczenia wynika, że zastosowanie go zdecydowanie polepsza warunki transmisji,



Rys. 3



Rys. 4

zwłaszcza przy współpracy z procesorami AVR. Rezystancja tego rezystora powinna wynosić około 3,3...4,7 kΩ, jednak przy podłączeniu większej liczby układów lub podłączeniu ich za pomocą dłuższych, kilkumetrowych przewodów, można wartość rezystancji zmniejszyć do około 1 kΩ.

Drugim popularnym układem z rodziny DS1820 jest układ oznaczony symbolem DS18B20. Sposób transmisji oraz podłączenia do mikrokontrolera są identyczne, a różnice polegają na sposobie odczytu temperatury oraz dokładności. Układ DS1820 zapisuje temperaturę z rozdzielczością 9 bitów (zakres od -55°C do 125°C, z rozdzielczością co 0,5°C), natomiast DS18B20 zapisuje temperaturę z rozdzielczością od 9 do 12 bitów.

Aby odczytać temperaturę z układu DS1820, należy wykonać kilka czynności związanych ze specyfiką magistrali 1Wire. Po pierwsze, musimy znać numer seryjny układu. Numer jest 8-bajtowy – zapisany na stałe w układzie. Numer ten ma następującą strukturę (rys. 4):

- Pierwszy odczytany bajt to numer rodziny (dla DS1820 to 10h, czyli 16 dziesiętnie, dla DS18B20 to 28h, czyli 40 dziesiętnie). Uwaga! Przy połączeniu tylko jednego układu można pominąć procedurę odczytu i zapisu nr seryjnego.

- Kolejne 6 bajtów to adres konkretnego egzemplarza układu.

- Ostatni bajt to suma kontrolna, którą w ramach tego artykułu nie będziemy się zajmować.

Jeżeli w systemie używamy jednego układu DS1820, to procedura odczytu przebiega następująco (listing 1):

- inicjalizacja magistrali,
- odczyt numerów seryjnych układów (w przypadku 1 układu ten krok może być pominięty),
- wybór konkretnego układu (w przypadku 1 układu ten krok może być pominięty),
- wysłanie żądania odczytu temperatury,
- oczekiwanie na konwersję temperatury,
- odczyt pierwszego bajtu danych,
- odczyt drugiego bajtu danych,
- ponowna inicjalizacja magistrali,
- programowa konwersja odczyta-

nej wartości na stopnie Celsjusza.

Magistralę 1Wire ustawiamy w stan gotowości do pracy poleceniem *1reset*. Jak można zauważyć, polecenie zaczyna się od *1w...* i jest regułą, że wszystkie polecenia dotyczące magistrali 1Wire zaimplementowane w Bascomie zaczynają się od *1w...*

W zależności od tego, ile układów dołączono do magistrali 1Wire, można zastosować dwa sposoby jej przeszukiwania. Przy dołączonym jednym układzie można od razu użyć funkcji *1wsearchfirst()*, która zwraca pierwszy znaleziony numer urzą-

dzenia z magistrali 1Wire. Przy dołączonych do magistrali kilku układach (jak na rys. 5) najlepiej użyć zestawu kilku funkcji, a mianowicie: *1wirecount()*, *1wsearchfirst()*, *1wsearchnext()*. Pierwsza z nich zwraca liczbę dostępnych układów, co można wykorzystać do identyfikacji dołączonych układów. Kolejne polecenie podaje numer seryjny pierwszego znalezionej układu. Ostatnie z wymienionych poleceń podaje numer następnego (kolejnego) układu. Wyboru docelowego układu 1Wire realizujemy, zapisując kolejno numer funkcji wyboru

List. 1. Program przeznaczony do współpracy z JEDNYM czujnikiem DS1820 podłączonym do magistrali 1Wire

```

$regfile = „m8def.dat”
$crystal = 4000000
Config 1wire = Portb.0                               'konfiguracja pinów

Declare Sub Pomiar

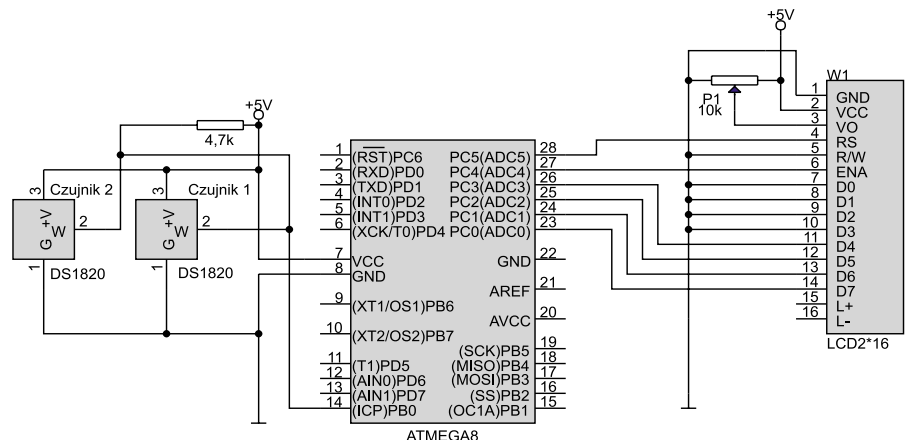
Deflcdchar 0 , 6 , 9 , 9 , 6 , 32 , 32 , 32 , 32       'deklaracja symbolu stopień

Dim T(2) As Byte                                     'tablica dwóch bajtów do przechowywania
                                                    'temperatury

Cls

Do
  Call Pomiar                                       'wywołanie procedury pomiar
  Cls                                               'ustaw kursor
  Lcd „Temp=” ;
  If T(2) > 0 Then Lcd „-” ;                       'wyswietlanie w zależności od znaku temperatury
  Lcd T(1) ;
  Lcd Chr(0) ; „C”
Loop
End

Sub Pomiar
  1wreset                                           'inicjacja magistrali 1WIRE
  If Err = 1 Then
    Cls                                             'kontrola błędu - jeśli err=1 wtedy
                                                    'nie znaleziono czujnika
    Lcd „Brak DS1820 „
    Do
      1wreset
      Loop Until Err = 0
    End If
    Lwwrite &HCC                                     'Wybór układu (dla 1 sztuki pomijamy wysyłanie
                                                    'numeru)
    Lwwrite &H44                                     'Polecenie konwersji temperatury
    Waitms 750                                     'czas konwersji 750 ms
    1wreset
    Lwwrite &HCC                                     'inicjacja magistrali 1WIRE
    Lwwrite &HCC                                     'Wybór układu (dla 1 sztuki pomijamy wysyłanie
                                                    'numeru)
    Lwwrite &HBE                                     'żądanie odczytania temperatury
    T(1) = 1wread()                                'odczytaj z magistrali 1WIRE 1 bajt
    T(2) = 1wread()                                'odczytaj z magistrali 1WIRE 2 bajt
    Lwreset
    If T(2) > 0 Then T(1) = 256 - T(1)              'do 1 elementu tablicy T
                                                    'do 2 elementu tablicy T
    Lwreset
    If T(2) > 0 Then T(1) = 256 - T(1)              'inicjacja magistrali 1WIRE
                                                    'Jeśli t(2)>0 to temperatura jest ujemna
    T(1) = T(1) / 2                                'oblicz temperaturę zgodnie z danymi noty
                                                    'katalogowej
  End Sub
    
```



Rys. 5

układu (tzw. Match ROM – kod polecenia 55h) oraz numer seryjny układu odczytany wcześniej:

```
lwwrite &H55
For I = 1 To 8
    lwwrite nr_ukladu(i)
Next I
```

Przy kilku układach dołączonych do magistrali należy wywoływać tę funkcję zawsze przed kolejnymi czynnościami (ponieważ dzięki temu żądane operacje przeprowadzimy tylko na wybranym układzie). Kolejny punkt, czyli wysłanie żądania odczytu temperatury, wykonujemy wysyłając do układu polecenie o kodzie 44h.

Następnym ważnym krokiem, który często powoduje frustrację użytkowników, jest oczekiwanie na konwersję temperatury. Po wysłaniu bajtu 44h układ DS1820 musi odczytać i przekonwertować dane do postaci dwóch liczb typu *byte*. Jeśli czas oczekiwania na wynik jest zbyt krótki, scalony termometr podaje jako wynik wartość „85”. W zależności od rozdzielczości oraz układu konieczne czasy oczekiwania są różne, najlepiej dobrać je eksperymentalnie. Jako wartość bezpieczną dla wszystkich układów z rodziny DS18xx można przyjąć 750µs.

Po dokonaniu konwersji możemy przystąpić do odczytu wartości temperatury. Wykonujemy to w sposób opisany wcześniej – wybieramy konkretny układ docelowy i następnie wysyłamy do układu wartość BEh:

```
lwwrite &HBE
'żądanie odczytania temperatury
T(1) = lwread()
'odczytaj z magistrali 1WIRE 1 bajt
T(2) = lwread()
'odczytaj z magistrali 1WIRE 2 bajt
```

Po wykonaniu tych czynności niezbędna jest ponowna inicjalizacja magistrali poleceniem *lwreset*.

Do wykonania pozostało ostatnie zadanie, czyli konwersja odczytanej wartości na stopnie Celsjusza. Jej przebieg jest uzależniony od typu układu, ponieważ – jak wspomnieliśmy wcześniej – DS1820 udostępnia zapis temperatury na 9 bajtach, a DS18B20 na 12 bajtach. W przypadku DS1820 konwersja temperatury z dokładnością 1 stopnia wygląda następująco:

```
If T(2) > 0 Then T(1) = 256 - T(1)
'Jeśli t(2)=0 to temperatura jest
'ujemna
T(1) = T(1)/2
'tu pozbywamy się ostatniego bitu
'części ułamkowej)
```

Pozostaje nam tylko wyświetlić temperaturę na wyświetlaczu:

```
Cls
Locate 1, 1 'ustaw kursor
Lcd „Temp=” ;
If T(2) > 0 Then Lcd „-“ ;
'wyświetlanie w zależności od znaku
```

List. 2 Procedura konwersji wyniku pomiaru

```
$regfile = „m8def.dat”
'XTAL
$crystal = 4000000
Config lwire = Portb.0
Declare Sub Pomiar
Declare Sub Wyb_uklad

Dim Nr_ukladu1(8) As Byte
Dim Nr_ukladu2(8) As Byte
Dim I As Byte, A As Byte
Dim T(2) As Byte
Dim nika wartosci
Dim Temp As Integer
Dim Ułamek As Integer
Dim Znak As Byte

Defodchar 0, 12, 18, 18, 12, 32, 32, 32, 32

Cls
Nr_ukladu1(1) = lwsearchfirst()
Nr_ukladu2(1) = lwsearchnext()

Do
    For A = 1 To 2
        If A = 1 Then Locate 1, 1 Else Locate 2, 1

        Call Pomiar
        Lcd „T(„ ; A ; „)=“ ;

        If Znak = 1 Then Lcd „-“ ;
        Lcd Temp ; „.” ;
        If Ułamek = 625 Then Lcd „0“ ;
        Lcd Ułamek ; Chr(0) ; „C”

        Next A
    Loop

Sub Pomiar
    lwreset
    If Err = 1 Then
        odpowiada do err=1 i
        Cls
        Lcd „Brak układu”
        Do
            lwreset
        Loop Until Err = 0

    End If
    Call Wyb_uklad
    lwwrite &H44
    Waitms 750

    lwreset
    Call Wyb_uklad
    lwwrite &HBE
    T(1) = lwread()
    T(2) = lwread()
    lwreset

'Wynik w DS18B20 jest zapisywany w dwóch bajtach. High=SSSS SHHH, Low = HHHH LLLL,
'gdzie S=znak, H-bity czesci calkowitej pomiaru temperatury w stopniach [C],
'L-bity wyniku po przecinku (L*0,0625 [C]).
'czyli najlepiej trzeba zlaczyz w jeden bajt bity SHHH oraz HHHH co da temperature zapisana wprost
'na 7 bajtach (H) - jezeli bedzie ujemna wtedy S = 1 i nalezy ja przeliczyz: 255-0HHHHHH
'Dodatkowo nalezy z bajtu Low usunac czesc HHHH i pozostanie 0000LLLL czyli czesc ułamkowa
'to co pozostanie nalezy pomnozyc przez 0,0625 s (rozdzielczosc maksymalna)
'procedura ponizsza jest niewersalna i dziala dobrze dla kazdej ustawionej rozdzielczosci czujnika
'od 9 do 12 bitow

    Temp = Makeint(t(1), T(2))

    Temp = Temp / 16

If T(2) = &HFF And T(1) > &HF0 Then Znak = 1

    T(1) = T(1) And &H0F
    Ułamek = T(1) * 625

    If T(2) > 127 Then Ułamek = 10000 - Ułamek

End Sub

Sub Wyb_uklad
    lwwrite &H55
    For I = 1 To 8
        If A = 1 Then lwwrite Nr_ukladu1(i)
        If A = 2 Then lwwrite Nr_ukladu2(i)
    Next I
End Sub
```

```
'temperatura
Lcd T(1) ;
Lcd Chr(0) ; „C”
'wyświetlenie znaku stopień Celsjusza
```

Niektóre źródła podają inny sposób wyświetlania temperatury odczytanej z czujnika, mianowicie wykorzystuje się polecenie *fusing* oraz zmienną typu *single*:

```
Temperatura = Fusing(temp, „#.“)
```

Jest to polecenie wysoce nieefektyw-

ne, które w testowanym układzie zwiększało rozmiar pamięci programu o ponad 500 bajtów, co w wielu przypadkach (np. przy zastosowaniu popularnego AT90S2313 stanowi ponad 25% jego pamięci) dyskwalifikuje je.

Przedstawimy także dwie procedury związane z obsługą czujnika DS18B20. Tematami, które zazwyczaj są pomijane lub wygłasza się o nich błędne opi-

nie, jest zmiana rozdzielczości pomiaru oraz obsługa czujników w trybach niższej rozdzielczości. W nocie katalogowej układu DS18B20 znajduje się informacja, że jeśli wybrano tryb konwersji 12-bitowy, to ułamek jest zapisany na 4 bitach, jeśli 11-bitowy, to ułamek zapisano na 3 bitach, w przypadku 10-bitowego trybu pomiaru ułamek zakodowano na 2 bitach, a w przypadku pomiaru 9-bitowego ułamek znajduje się na jednym bicie (czyli 3 najmłodsze bity są nieistotne).

Podstawową procedurę konwersji pokazano na **listingu 2**. Jest to procedura dosyć często spotykana w różnych poradach (ale często bywa w nich niekompletna informacja, ograniczająca się do stwierdzenia „Podziel przez 16, zamiast przez 2, jak dla DS1820”). W przykładzie przedstawiono sposób odczytu włącznie z usunięciem ograniczeń wprowadzanych poprzez zastosowanie zmiennej *Integer* (problem ze znakiem – przy temperaturach wyższych niż -1 stopień, w zakresie od -1 do zera).

Przedstawię jeszcze alternatywny sposób obliczania i wyświetlania temperatury. Jego zaletą jest fakt, że program po kompilacji jest krótszy od tego w przykładzie o 90 bajtów.

Wynik pomiaru DS18B20 jest zapisywany w dwóch bajtach:

```
High=SSSS SHHH,
Low = HHHH LLLL,
gdzie:
```

S – znak,

H – bity części całkowitej pomiaru temperatury w stopniach [C],

L – bity wyniku po przecinku (L*0,0625 [C]).

Po odczytaniu danych z DS1820, należy złączyć w jeden bajt bity SHHH oraz HHHH, co da wartość temperatury zapisaną na 7 bitach (H). Jeśli temperatura będzie ujemna, wtedy S = 1 i należy ją przeliczyć zgodnie ze wzorem:

```
255-0HHHHHHH
```

Dodatkowo z bajtu *Low* należy usunąć część HHHH tak, że pozostanie 0000LLLL, czyli część ułamkowa. To co pozostanie, należy pomnożyć przez 0,0625 (rozdzielczość maksymalna):

```
I = T(1)
'tu wykorzystujemy zmienna I jako
'zmienna pomocnicza przy wyświetlaniu
'części ułamkowej
I = I And &H0F
'tu pozbywamy się 4 niepotrzebnych
'bitów (część całkowita) - zostaje
'tylko część ułamkowa
Shift T(1), Right, 4
'tu pozbywamy się bitów ułamka
'części T(1) ponieważ mamy je już
'odseparowane w I
Shift T(2), Left, 4
'a tu „wycinamy” niepotrzebne nam
'4 początkowe bity znaku (zostaje
'tylko piąty z nich jako najstarszy
'w części t(2))
```

```
T(1) = T(2) Or T(1)
'łączmy skonwertowane t(1) i t (2)
'w zmienną T(1)
Ułamek = I * 625
'mnożymy razy maksymalną rozdzielczość
If T(1) > 127 Then
'jeśli t(2) mamy > 127 to znaczy
'że bit znaku jest ustawiony czyli
'liczba jest ujemna
T(1) = 255 - T(1)
'konwertujemy temperaturę ujemną
Ułamek = 10000 - Ułamek
End If
```

Użytkownikowi pozostaje sprawdzić podczas wyświetlania znak wyniku oraz skontrolować, czy część ułamkowa nie jest równa 625 – wtedy należy wyświetlić jeszcze przed zmienną *Ułamek* znak „0”.

Na **listingu 2** pokazano alternatywny sposób odczytu temperatury z czujnika, przy wykorzystaniu zmiennej temp typu *integer* (co powoduje wygenerowanie dłuższego kodu wynikowego). Do przechowywania numerów układów wykorzystujemy tablice *nr_ukladu(8)* i *nr_ukladu-2(8)*. Numer układu, z którego będziemy czytać dane, przepisujemy z wybranej tablicy do tablicy *Biezacy(8)*. W procedurze *Wybierz_uklad* dokonujemy zapisu numeru układu do magistrali, natomiast w pętli głównej w zmiennej *A* przechowujemy numer indeksu aktualnie odczytywanego układu. W celu oszczędzenia pamięci można wykorzystać pamięć EEPROM do przechowywania numerów układów – przykład znajduje się na stronie EP w dziale *Download* oraz na płycie CD-EP11/2004B.

Teraz przedstawimy sposób przedstawienia czujnika DS18B20 w tryb mniejszej rozdzielczości. Wskazane jest to wtedy, kiedy nie zależy nam na dokładności, ale na czasie odczytu i konwersji – w trybie 12-bitowym czas konwersji jest niemal 8 razy dłuższy (750 ms) niż w przypadku trybu 9-bitowego (100 ms):

```
lwreset
Call Wybierz_uklad
lwwrite &H4E
'Polecenie zapisu pamięci
lwwrite &H00
'Zapis dowolnego bajtu
'(pamięć użytkownika)
lwwrite &H00
'Zapis dowolnego bajtu
'(pamięć użytkownika)
lwwrite &B00011111
'Zapis bajtu konfiguracyjnego
'9 bitów
'lwwrite &B00111111
'Zapis bajtu konfiguracyjnego 10 bitów
'lwwrite &B01011111
'Zapis bajtu konfiguracyjnego 11 bitów
'lwwrite &B01111111
'Zapis bajtu konfiguracyjnego 12 bitów
lwreset 'koniec
```

Po wykonaniu procedury przełączającej układ jest ustawiany w tryb 9-bitowy, który utrzymuje się do wyłączenia zasilania (konfiguracja nie jest zachowywana na stałe). Po ponownym włączeniu zasilania układ startuje w trybie 12-bitowym. Aby zapisać na stałe

konfigurację w trybie 9-bitowym, należy wykonać komendę *1wwrite* (z argumentem 48h), która kopiuje zawartość bitów konfiguracyjnych do pamięci nieulotnej EEPROM:

Na koniec przedstawiamy procedurę odczytu temperatury z czujnika DS1820 z rozdzielczością 0,1°C. Wykorzystujemy w tym celu dane zawarte w sumie kontrolnej oraz zastrzeżonych zakresach pamięci. Wykorzystuje ona odczytaną tablicę całej pamięci oraz pewne modyfikacje:

```
Dim T As Integer
'tablica dwóch bajtów
'do przechowywania temperatury
Dim Bd(9) As Byte
'tablica dwóch bajtów
'do przechowywania temperatury
Dim Tmp As Byte
'tablica dwóch bajtów
'do przechowywania temperatury
Dim T1 As Byte
(.....)

lwwrite &HCC
'Wybór układu (dla 1 sztuki
'pomijamy wysyłanie numeru)
lwwrite &H44
'Polecenie konwersji temperatury
Waitms 750
'czas konwersji 750 ms
lwreset
'inicjacja magistrali 1WIRE
lwwrite &HCC
'przeskocz zapis numeru ID
lwwrite &HBE
'rozkaz odczytu danych
Bd(1) = lwread(9)
'odczyt danych do tablicy bd
lwreset
'zerowanie magistrali 1wire

'kolejne instrukcje obliczają
'temperaturę z precyzją 0,1 st C
Tmp = Bd(1) And 1
'obliczenie wartości zmiennej tmp
If Tmp = 1 Then Decr Bd(1)
'jeśli tmp=1 to zmniejsz o jeden
'pierwszy element tablicy bd
T = Makeint(bd(1) Bd(2))
'łączy dwa bajty w zmienna integer
T = T * 50
T = T - 25
T1 = Bd(8) - Bd(7)
'obliczenie różnicy zastrzeżonych
'bajtów
T1 = T1 * 100
'obliczenie temperatury
T1 = T1/Bd(8)
T = T + T1
T = T/10
'obliczenie temperatury (zmienna T
'zawiera obliczana temperaturę)

Cls
Lcd „Temp=” ; T
```

Chcę podkreślić, że ze względu na objętość procedury i jej „pamięciożerność” tam, gdzie to jest możliwe należy stosować czujniki DS18B20. Na stronie EP w dziale *Download* umieszczono dodatkowe pliki, w których pokazano stosowanie różnych procedur odczytu oraz sposób korzystania z różnych czujników w jednym programie (DS1820 i DS18B20), sposób korzystania z pamięci EEPROM do przechowywania numerów czujników oraz przykłady wykorzystane w artykule.

Artur Starz, EP
artur.starz@ep.com.pl