

LITEcomp – aplikacje

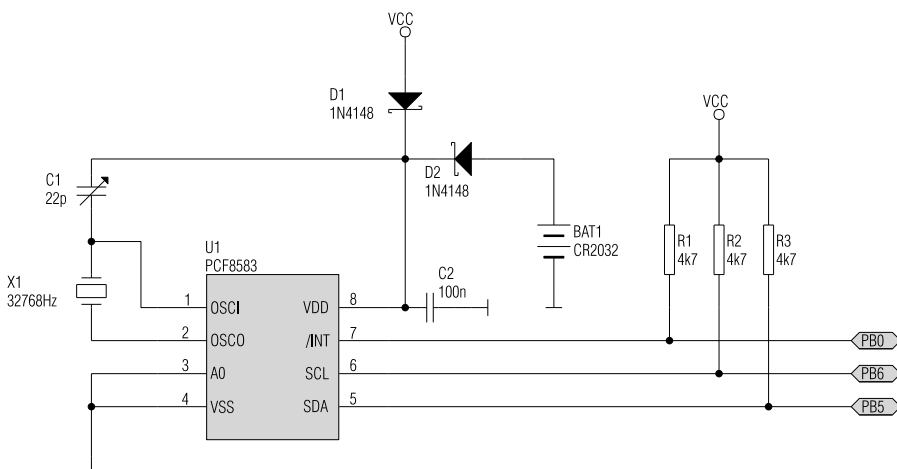
Zegar z układem PCF8583

W artykule przedstawiamy zastosowanie komputerka LITEcomp do budowy zegara z tanim i łatwo dostępnym układem PCF8583. Na przykładzie tego układu pokazemy w jaki sposób obsłużyć magistralę I²C za pomocą mikrokontrolerów z rodziny ST7LITE.

Bazą układu zegara jest moduł LITEcomp z układem PCF8583 dołączonym według schematu przedstawionego na rys. 1. Do podłączenia układu wykorzystano trzy wyprowadzenia I/O modułu LITEcomp: dwa są przeznaczone do obsługi magistrali I²C, natomiast trzecie jest wejściem przerwania generowanego co sekundę przez układ PCF8583. Odczyt czasu oraz jego wyświetlenie odbywa się właśnie w ramach obsługi przerwania generowanego na wyprowadzeniu INT układu PCF8583. Pozostały czas mikrokontroler spędza w pętli sprawdzającej stan przycisków S1 i S2 modułu LITEcomp

Dla praktyków

Zachęcamy Czytelników do przysyłania własnych opracowań na komputerku LITEcomp. Najlepsze opisy opublikujemy w EP, wszystkie nadesłane projekty uhonorujemy interesującymi nagrodami (książki, zestawy uruchomieniowe, narzędzia – wykaz opublikujemy za miesiąc).



Rys. 1.

służących do wprowadzania nowych wartości czasu oraz daty.

Układ zegara PCF8583 jest na tyle dobrze znany, że pozwoliłem sobie zrezygnować z dokładnego opisywania jego funkcji, które są opisane zarówno w dokumentacji układu, jak również w archiwalnych numerach Elektroniki Praktycznej. Funkcjonalność przedstawianego układu zegara ogranicza się do wyświetlenia na wyświetlaczu LCD modułu LITEcomp aktualnego czasu i daty oraz nastawieniu ich wartości za pomocą przycisków modułu LITEcomp. Nic nie stoi na przeszkodzie, aby poszerzyć układ o dodatkowe funkcje, zwłaszcza, że obecny program zajmuje niecałe 25% dostępnej pamięci Flash mikrokontrolera ST7FLITE19.

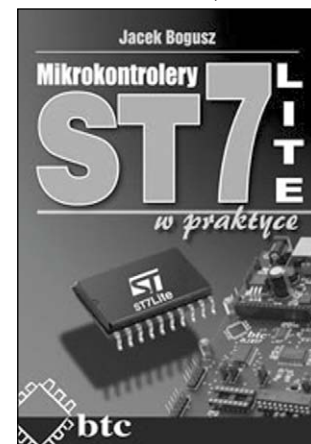
Zrobione

ST7LITE
na LITEcomp
st7.ep.com.pl

Dołączony materiał do artykułu
publikujemy na CD-EP oraz www.ep.com.pl

LITEcomp

LITEcomp jest prostym komputerkiem wykonanym na mikrokontrolerze ST7FLITE19. LITEcomp jest w ramach promocji dodawany bezpłatnie do książki „Mikrokontrolery ST7LITE w praktyce” (autor Jacek Bogusz). Książka jest dostępna w sklep.avt.pl (numer katalogowy KS-260905).



Obsługa magistrali I²C

Na rys. 2 „streszczono” zasadę działania magistrali I²C. Do jej obsługi wykorzystano wyprowadzenia 5 i 6 portu B. Konieczne jest zastosowanie zewnętrznych rezystorów podciągających

o rezystancji ok. 4,7 kΩ na liniach SDA i SCL. W razie potrzeby można wykorzystać inne wyprowadzenia mikrokontrolera zmieniając odpowiednio wartości masek odpowiedzialnych za modyfikację stanów linii SDA i SCL. Sterowanie stanem linii SDA i SCL zostało zrealizowane poprzez modyfikację stanu odpowiednich bitów w rejestrze kierunku portu. Metoda ta pozwala na zwolnienie programisty z obowiązku przełączania kierunku pracy portu w przypadku klasycznego sterowania stanem wyprowadzeń za pomocą rejestru danych PBDR. Stan wysoki na linii jest wymuszany przez zewnętrzny rezystor podciągający przy wyprowadzeniu skonfigurowanym jako wejście „pływające” (wyzerowany bit w rejestrze kierunku). Natomiast stan niski jest wymuszany poprzez wymuszenie przez wewnętrzny tranzystor portu skonfigurowanego w tryb wyjściowy z otwartym kolektorem (ustawiony bit w rejestrze kierunku). Modyfikacja stanu rejestru odbywa się poprzez wykonanie operacji iloczynu logicznego z maską zanegowaną (zerowanie bitu) bądź sumy logicznej z maską (ustawienie bitu). Wynikają z tego następujące wartości masek:

```
SDA_MASK EQU %00100000
SDA_MASK_INV EQU %11011111
SCL_MASK EQU %01000000
SCL_MASK_INV EQU %10111111
```

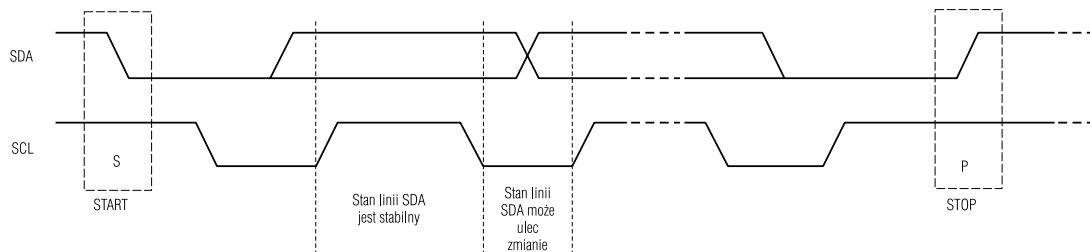
W celu ustawienia na wybranej linii magistrali odpowiedniego stanu przygotowałem makroinstrukcje przedstawione poniżej:

List. 1. Procedura generująca sekwencję START

```
.i2c_start
SET_SDA
SET_SCL
CLR_SDA
CALL i2c_delay
CLR_SCL
RET
```

List. 2. Procedura generująca sekwencję STOP

```
.i2c_stop
SET_SCL
CALL i2c_delay
SET_SDA
RET
```



Rys. 2.

```
SET_SDA_MACRO
LD A, PBDDR
AND A, #SDA_MASK_INV
LD PBDDR, A
MEND
```

```
CLR_SDA_MACRO
LD A, PBDDR
OR A, #SDA_MASK
LD PBDDR, A
MEND
```

```
SET_SCL_MACRO
LD A, PBDDR
AND A, #SCL_MASK_INV
LD PBDDR, A
MEND
```

```
CLR_SCL_MACRO
LD A, PBDDR
OR A, #SCL_MASK
LD PBDDR, A
MEND
```

Procedura generująca sekwencję START

Wygenerowanie sekwencji START polega na wymuszeniu opadającego zbocza na linii SDA przy wysokim stanie linii SCL. Kod procedury przedstawiono na list. 1.

List. 3. Procedura wysyłająca bajt danych na magistralę I²C

```
.i2c_write
LD i2c_data, A ; zapamiętanie A w zmiennej i2c_data
LD A, #8
LD i2c_counter, A ; ilość powtórzeń pętli
i2cw1
CLR_SCL ; stan niski na linii SCL
CLR_SDA ; stan niski na linii SDA
LD A, i2c_data ; przywrócenie danej
RLC A ; obrót w lewo przez C
LD i2c_data, A ; zapamiętanie danej
JRNC i2cw2 ; skok, jeśli C wyzerowany
SET_SDA ; w przeciwnym razie ustawienie linii SDA
i2cw2
SET_SCL ; ustawienie linii SCL
DEC i2c_counter ; zmniejszenie licznika pętli
JRNE i2cw1 ; skok do i2cw1 jeśli większe od zera
CLR_SCL ; wyzerowanie linii SCL
SET_SDA ; wysoki stan na linii SDA
SET_SCL ; wysoki stan na linii SCL
RCF ; wyzerowanie wskaźnika C
LD A, PBDR
AND A, #SDA_MASK ; porównanie stanu portu z maską
JREQ i2cw3 ; skok do i2cw3, jeśli równe
SCF ; w przeciwnym razie wyzerowanie wskaźnika C
i2cw3
CLR_SDA ; wyzerowanie linii SDA
CLR_SCL ; wyzerowanie linii SCL
RET
```

List. 4. Procedura odbierająca bajt danych z magistrali I²C

```
.i2c_read
RLC A ; zapamiętanie bitu ACK
LD i2c_data, A
LD A, #8
LD i2c_counter, A ; ilość powtórzeń pętli
SET_SDA ; stan wysoki na linii SDA
i2cr1
CALL i2c_delay ; czekamy chwilę
SET_SCL ; stan wysoki na linii SCL
RCF ; zerujemy wskaźnik C
LD A, PBDR ; odczytujemy stan portu
AND A, #SDA_MASK ; sprawdzamy stan linii SDA
JREQ i2cr2 ; jeśli wyzerowany to skok do i2cr2
SCF ; w przeciwnym razie ustawiamy wskaźnik C
i2cr2
LD A, i2c_data ; przywracamy poprzednio odczytane bity
RLC A ; zapamiętujemy nowo odebrany bit
LD i2c_data, A ; zapamiętujemy
CLR_SCL ; niski stan na linii SCL
DEC i2c_counter ; zmniejszamy licznik pętli
JRNE i2cr1 ; skok do i2cr1 jeśli większe od zera
CLR_SDA ; zerujemy linię SDA
JRNC i2cr3 ; skok do i2cr3 jeśli wskaźnik C wyzerowany
SET_SDA ; ustawiamy linię SDA
i2cr3
SET_SCL ; ustawiamy linię SCL
CLR_SCL ; zerujemy linię SCL
CLR_SDA ; zerujemy linię SDA
LD A, i2c_data ; do A ładujemy odebrana dana
RET
```

List. 5. Procedura obsługi wyświetlacza LCD

```
; Procedura wyłączająca miganie kursora
.cursoroff
LD A, #00001100
CALL writecommand
RET
; Procedura włączająca miganie kursora
.cursoron
LD A, #00001111
CALL writecommand
RET
; Procedura powrotu kursora na początek wyświetlacza
.lcdhome
LD A, #00000010
CALL writecommand
RET
; Procedura ustawiająca adres, pod który zapisywane będą dane
.lcdsetaddress
OR A, #10000000
CALL writecommand
RET
```

List. 6. Procedura odczytująca czas i datę z układu PCF8583

```
.readtime
CALL 2c_start ; start transmisji
LD A, #0A0 ;
CALL i2c_write ; zapis adresu układu PCF8583
LD A, #02 ;
CALL i2c_write ; zapis adresu rejestru wewnętrznego
CALL i2c_start ; ponowny start
LD A, #0A1 ; zapis adresu układu PCF8583 + R
CALL i2c_write
RCF ; ACK
CALL i2c_read ; odczyt wartości sekund
LD sekundy, A
RCF ; ACK
CALL i2c_read ; odczyt wartości minut
LD minuty, A
RCF ; ACK
CALL i2c_read ; odczyt wartości godzin
LD godziny, A
RCF ; ACK
CALL i2c_read ; odczyt wartości daty
LD {packdate+0}, A
SCF ; NACK
CALL i2c_read ; odczyt wartości daty
LD {packdate+1}, A
CALL i2c_stop ; stop transmisji
RET
```

tru z układem DS18B20 (EP5/2007). W stosunku do poprzedniego projektu zostały dokonane pewne modyfikacje w metodzie zmiany stanu wyprowadzeń E i RS, eliminujące użycie instrukcji BSET/BRES. Dodano również pewne pomocnicze procedury, których kod przedstawiono na list. 5.

Procedura obsługi przerwania zewnętrznego

Odczyt i wyświetlenie aktualnego czasu zrealizowane jest procedurze obsługi przerwania zewnętrznego generowanego co 1 sekundę przez układ PCF8583 na wyjściu INT, które jest podłączone do wyprowadzenia PBO mikrokontrolera. Wyprowadzenie to jest skonfigurowane do reakcji na opadające zboczne sygnału zewnętrznego. Wyprowadzenie PBO należy do grupy wyprowadzeń mogących generować sygnał przerwania EI0.

Procedurę obsługi przerwania zewnętrznego EI0 przedstawiono poniżej:

```
.ei3_vect
CALL readtime ; odczyt czasu
i date
CALL disptime ; wyświetlenie
czasu i daty
IRET
```

Sekwencję odczytu przedstawiono na rys. 3. Każdorazowy odczyt bajtu powoduje automatyczną inkrementację adresu rejestru wewnętrznego, dzięki czemu cała operacja odczytu zajmuje znacznie mniej czasu niż w przypadku adresowania każdego rejestru z osobna. Kod procedury odczytującej czas i datę z układu PCF8583 przedstawiono na list. 6.

Ponieważ w rejestrach układu PCF8583 czas oraz data są przechowywane w formacie BCD, konieczne jest przygotowanie procedury umożliwiającej wyświetlenie liczby w kodzie BCD na wyświetlaczu LCD. Kod procedury wyświetlającej liczbę BCD przedstawiono na list. 7.

Informacja o aktualnej dacie jest przechowywana w postaci upakowanej. Dlatego też przygotowano

Procedura generująca sekwencję STOP

Wygenerowanie sekwencji STOP polega na wymuszeniu narastającego zbocza na linii SDA przy wysokim stanie linii SCL. Kod procedury jest przedstawiony na list. 2.

Procedura wysyłająca bajt danych na magistralę I²C

Zadaniem tej procedury (list. 3) jest wysłanie na magistralę bajtu danych oraz odebranie nadanego przez układ podrzędny bitu potwierdzenia ACK. Odebrany bit

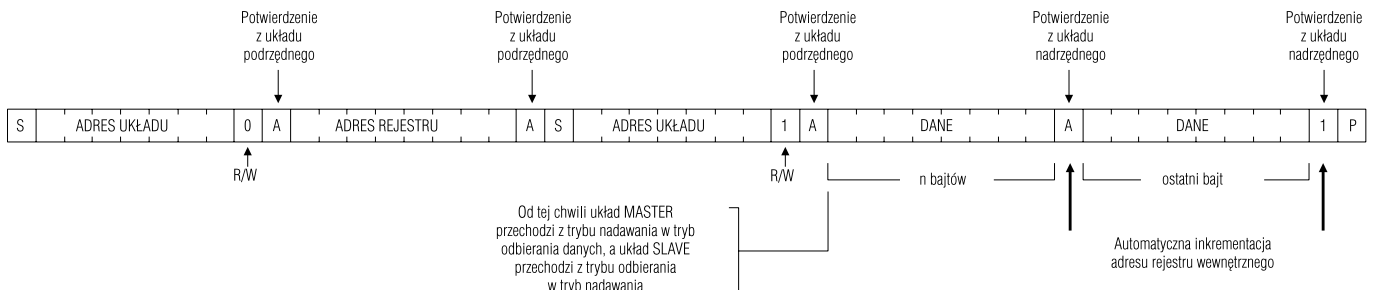
ACK zwracany jest we wskaźniku C.

Procedura odbierająca bajt danych z magistrali I²C

Zadaniem procedury z list. 4 jest odebranie bajtu danych z magistrali I²C oraz nadanie bitu potwierdzenia ACK. Wartość bitu ACK należy przed wywołaniem procedury umieścić we wskaźniku C.

Obsługa wyświetlacza LCD

Procedury obsługi wyświetlacza LCD są podobne do przedstawionych w artykule dotyczącym termome-



Rys. 3.

procedurę dokonującą rozpakowania informacji o dacie na dni i miesiące oraz dni tygodnia w formacie BCD. Kod procedury przedstawiony jest na **list. 8**.

Wyświetlanie aktualnego czasu i daty

Po rozpakowaniu daty oraz przygotowaniu procedury umożliwiającej wyświetlenie liczby BCD pozostaje nam już tylko wyświetlić odczytaną z układu PCF8583 datę oraz czas. Procedurę wyświetlającą aktualny czas oraz datę przedstawiono na **list. 9**.

Ustawianie czasu

Wprowadzania wartości czasu oraz daty odbywa się za pomocą dwóch przycisków modułu LITEcomp. Zarówno wejście w tryb ustawiania, jak i przesuwanie kursora po poszczególnych pozycjach następuje po naciśnięciu przycisku S1. Inkrementacja wartości wskazanej przez kursor pozycji następuje po naciśnięciu przycisku S2. Czas i data przechowywane są w 16-bajtowej tablicy. Organizacja tej tablicy dokładnie odpowiada formatowi wyświetlanego czasu oraz daty na wyświetlaczu LCD. Każda komórka tablicy odpowiada znakowi na wyświetlaczu, jednak nie każda jej komórka jest odwzorowana na wyświetlaczu. Pozwoliłem sobie na znaczne uproszczenie – kursor przesuwa się po wszystkich szesnastu kolejnych pozycjach wyświetlacza i zarazem pozycjach tablicy przechowującej informacje o wprowadzanym czasie i dacie. Inkrementacja pozycji nie będącej składową czasu bądź daty powoduje inkrementację komórki zmiennej o odpowiedniej pozycji, jednakże zmiany te nie są wyświetlane na wyświetlaczu. Procedurę inkrementującą czas przedstawiono na **list. 10**.

Zapis ustawionego czasu do układu PCF8583

Po wprowadzeniu nowej wartości czasu i daty należy zapisać je do układu PCF8583. Jak już wiadomo, w rejestrach układu PC8583 informacja przechowywana jest w formacie BCD, przed zapisaniem należy więc dokonać konwersji z formatu naszej „struktury” przechowującej wprowadzony czas na postać BCD. W tym celu przygotowałem procedurę pokazaną na **list. 11**.

Po spakowaniu daty i czasu do postaci BCD można dokonać ich zapisu do układu PCF8583. Sekwencję

List. 7. Procedura wyświetlająca liczbę BCD na LCD

```
.writebcd
PUSH A ; zapamiętujemy liczbę na stosie
AND A, #$F0 ; zerujemy młodszy półbajt
SWAP A ; zamiana półbajtów miejscami
ADD A, #'0' ; dodajemy kod znaku ,0' (konwersja na kod ASCII)
CALL writedata ; wyświetlamy na wyświetlaczu

POP A ; odtwarzamy liczbę ze stosu
AND A, #$0F ; zerujemy starszy półbajt
ADD A, #'0' ; dodajemy kod znaku ,0' (konwersja na kod ASCII)
CALL writedata ; wyświetlamy na wyświetlaczu
RET
```

List. 8. Procedura rozpakowująca informacje o dacie na dni i miesiące oraz dni tygodnia do formatu BCD

```
.unpackdate
LD A, {packdate+0} ; załadowanie do A pierwszego bajtu spakowanej daty
AND A, #%00111111 ; wyzerowanie numeru roku
LD dni, A ; zapis wartości dnia

LD A, {packdate+1} ; załadowanie do A drugiego bajtu spakowanej daty
AND A, #%00011111 ; wyzerowanie dnia tygodnia
LD miesiace, A ; zapis wartości miesiąca

LD A, {packdate+1} ; załadowanie do A drugiego bajtu spakowanej daty
SWAP A ; zamiana półbajtów
SRA A ; przesunięcie o 1 bit w prawo
AND A, #%00000111 ; wyzerowanie nieznaczących bitów
LD dnityg, A ; zapis wartości dnia tygodnia
RET
```

List. 9. Procedura wyświetlająca czas oraz datę

```
.disptime
CALL lcdhome
LD A, godziny
CALL writebcd ; wyświetlamy godziny
LD A, #' :'
CALL writedata ; wyświetlamy dwukropek
LD A, minuty
CALL writebcd ; wyświetlamy minuty
LD A, #' :'
CALL writedata ; wyświetlamy dwukropek
LD A, sekundy
CALL writebcd ; wyświetlamy sekundy
CALL unpackdate ; rozpakowanie daty
LD A, #DATE_HOME
CALL lcdsetaddress ; ustawienie adresu
LD A, dni
CALL writebcd ; wyświetlenie dnia
LD A, #' / '
CALL writedata ; wyświetlenie ukośnika
LD A, miesiace
CALL writebcd ; wyświetlenie miesiąca
RET
```

List. 10. Procedura inkrementująca nastawę czasu

```
.inctemptime
LD X, cursor ; załadowanie do X pozycji kursora
INC (czas,X) ; inkrementacja komórki pamięci wskazanej przez kursor
LD A, (czas,X)
CP A, #10 ; sprawdzenie czy > 9
JRNE inctt1 ; jeśli nie to skok do inctt1
CLR (czas,X) ; w przeciwnym razie wyzerowanie
inctt1
CALL cursoroff ; wyłączenie wyświetlania kursora
CALL disptemptime ; wyświetlenie ustawianego czasu
CALL setcursor ; ustawienie kursora
CALL cursoron ; włączenie kursora
RET
```

zapisu danych do układu PCF8583 przedstawiono na **rys. 4**. Podczas zapisu wskaźnik adresu rejestru wewnętrzznego układu PCF8583 jest inkrementowany automatycznie. Przed zapisaniem rejestrów daty i czasu następuje zatrzymanie odliczania czasu, które zostaje włączone po przesłaniu do układu PCF8583 wszystkich nowostawionych wartości.

Kod procedury zapisującej czas i datę do układu PCF8583 przedstawiono na **list. 12**.

Po zapisaniu wprowadzonej daty i czasu do układu PCF8583 następuje odblokowanie przerwania zewnętrznego oraz powrót do pętli sprawdzającej stan przycisków S1 i S2.

Przedstawiony program pełni jedynie kilka niezbędnych funkcji, jakich wymaga się od zegara. W programie nie została zaimplementowana funkcja alarmu oraz inne funkcje występujące w zegarach, jak chociażby wyświetlanie nazwy dnia tygodnia. Program można potraktować jako bazę do włą-

Nie przegap!

interesujących materiałów w czasopiśmie

6 2007
ELEKTRONIKA dla wszystkich
www.eportal.pl

Niezwykły Oscyloskop cyfrowy
Uniwersalny sterownik wzmacniacza audio
EdE - Inteligentny symulator alarmu

www.eportal.pl

W czerwcowym, numerze **Elektroniki dla Wszystkich** m.in.:

■ **Niezwykły oscyloskop cyfrowy**
Jak wiadomo, nawet najprostszy oscyloskop może zaoszczędzić mnóstwo czasu podczas uruchamiania lub naprawy różnych urządzeń. Wiele parametrów elektrycznych da się zmierzyć różnymi miernikami, lecz aby zobaczyć kształt sygnału, niezbędny staje się wtedy opisany w artykule oscyloskop.

■ **Softstart do toroida - wersja de Luxe**
Transformatory toroidalne o mocach przekraczających 200W potrafią sprawić bardzo szybkie niespodzianki: podczas włączania napięcia gwałtownie palą się lub „wyskakują” bezpieczniki nie tylko w urządzeniu, a często też w domowej instalacji energetycznej. „Lekarstwem” na te problemy jest opisane w artykule urządzenie.

■ **Minitransceiver CW/80m**
Ten minitransceiver to niesłychanie prosty układ bardzo małej mocy (tzw. QRPp), spełniający wymagania także licencjonowanego krótkofalowca. W połączeniu z anteną na pasmo 80m może wchodzić w skład wakacyjnego zestawu podczas terenowych zawodów QRPp/CW. Idealnie nadaje się do przeprowadzonych treningów i nauki telegrafii a nawet do pierwszych łączności na paśmie.

■ **Termometr elektroniczny**
Termometr to „żelazny” temat wśród projektów elektronicznych. Przedstawiony w artykule projekt wykaże się dużą przydatnością w każdym domu. Prezentowane rozwiązanie, poza pomiarem temperatury z rozdzielczością 0,1°C w zakresie -55°C do +125°C, stanowi też rejestrator najwyższej i najniższej temperatury.

Kolejny projekt dla zupełnie początkujących:
Inteligentny wskaźnik/symulator alarmu

- PONADTO W NUMERZE:**
- Uniwersalny sterownik wzmacniacza audio
 - Zwirowana linijka
 - Inteligentny czujnik zbliżeniowy IRED
 - Zamek elektroniczny na klucz częstotliwościowy
 - Ensave. Ciulacz energii
 - Kempingowy wiatraczek
 - Dwukolorowa lampka
 - Druga młodość Autotraxa - Trasowanie ścieżek
 - Pod lupą - Para różnicowa
 - TVSAT - DISEqC
 - Moc przebiegów zmiennych
 - Szkoła Konstruktorów - Uzupelnnienie zegara
 - Szkoła Konstruktorów - Magiczne napisy
 - Szkoła Konstruktorów - Wykorzystanie PCA-100
 - Inteligentny dom, czyli ofensywa optoelektroniki - Transmisja sygnałów

A może masz pomysł na ciekawy artykuł lub projekt? Skonstruowałeś urządzenie, które jest godne zaprezentowania szerszej publiczności? Możesz napisać artykuł edukacyjny? Chcesz podzielić się doświadczeniem?

W takim razie zapraszamy do współpracy na łamach „Elektroniki dla Wszystkich”. Kontakt: edw@eportal.pl.

EdW możesz zamówić w sklepie internetowym AVT
http://www.sklep.avt.com.pl, telefonicznie 022 568 99 50,
fax: 022 568 99 55, listownie lub za pomocą e-maila
handlowy@avt.com.pl. Do kupienia także w Empikach
i wszystkich większych kioskach z prasą.
Na wszelkie pytania czeka także Dział Prenumeraty
tel. 022 568 99 22, prenumerata@avt.com.pl.

List. 11. Procedura konwersji czasu z formatu naszej „struktury” przechowywanej wprowadzony czas na postać BCD

```
.packtime
LD A, {czas+0} ; do A ładujemy dziesiątki godzin
SWAP A ; zamiana półbajtów
OR A, {czas+1} ; suma logiczna z jednostkami godzin
LD godziny, A ; ładujemy do godzin

LD A, {czas+3} ; do A ładujemy dziesiątki minut
SWAP A ; zamiana półbajtów
OR A, {czas+4} ; suma logiczna z jednostkami minut
LD minuty, A ; ładujemy do minut

LD A, {czas+6} ; do A ładujemy dziesiątki sekund
SWAP A ; zamiana półbajtów
OR A, {czas+7} ; suma logiczna z jednostkami sekund
LD sekundy, A ; ładujemy do sekund

LD A, {czas+11} ; do A ładujemy dziesiątki dni
SWAP A ; zamiana półbajtów
OR A, {czas+12} ; suma logiczna z jednostkami dni
LD dni, A ; ładujemy do dni

LD A, {czas+14} ; do A ładujemy dziesiątki miesięcy
SWAP A ; zamiana półbajtów
OR A, {czas+15} ; suma logiczna z jednostkami miesięcy
LD miesiace, A ; ładujemy do miesięcy
RET
```

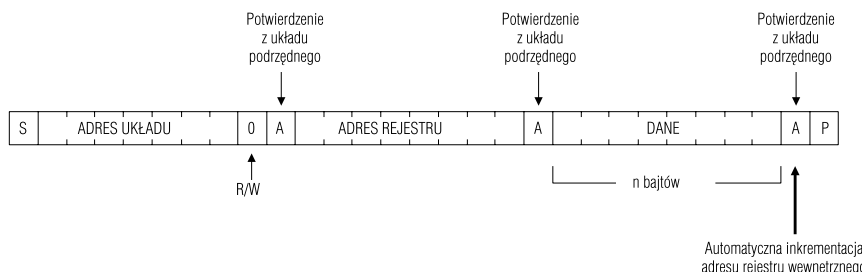
List. 12. Procedura zapisująca czas i datę do układu PCF8583

```
.writetime
CALL i2c_start ; start transmisji
LD A, #SA0 ; adres układu
CALL i2c_write ; zapis
LD A, #S00 ; adres rejestru wewnętrznego układu PCF8583
CALL i2c_write ; zapis
LD A, #S80 ; zatrzymanie odliczania czasu
CALL i2c_write ; zapis
LD A, #S00 ; z
CALL i2c_write ; zapis
LD A, sekundy ; zapis sekund
CALL i2c_write ; zapis minut
LD A, minuty ; zapis minut
CALL i2c_write ; zapis godzin
LD A, godziny ; zapis godzin
CALL i2c_write ; zapis dni
LD A, dni ; zapis dni
CALL i2c_write ; zapis dni
LD A, miesiace ; zapis miesięcy
CALL i2c_write ; zapis miesięcy
CALL i2c_stop ; stop
CALL i2c_start ; start
LD A, #SA0 ; adres układu
CALL i2c_write ; adres układu
LD A, #S00 ; adres rejestru wewnętrznego
CALL i2c_write ; włączenie odliczania czasu
LD A, #S00 ; włączenie odliczania czasu
CALL i2c_write ; stop
CALL i2c_stop ; stop
RET
```

nych eksperymentów prowadzących do stworzenia bardziej rozbudowanego układu zegara, którego dodatkowe funkcje użytkownik może zaimplementować samodzielnie i adekwatnie do własnych potrzeb. Program w przedstawionej w artykule postaci korzysta tylko z pierwszej linii alfanumerycznego wyświetlacza LCD, wyświetlając w drugiej stały tekst, jakim jest adres strony internetowej Elektroniki

Praktycznej. Tekst ten może zostać dowolnie zmieniony przez użytkownika. Drugą linię wyświetlacza można również dowolnie wykorzystać przy dodawaniu własnych funkcji - można się pokusić na przykład o połączenie obecnego programu z poprzednim i stworzenie zegarka wyświetlającego temperaturę otoczenia.

Radosław Kwiecień, EP
radoslaw.kwiecien@ep.com.pl



Rys. 4.