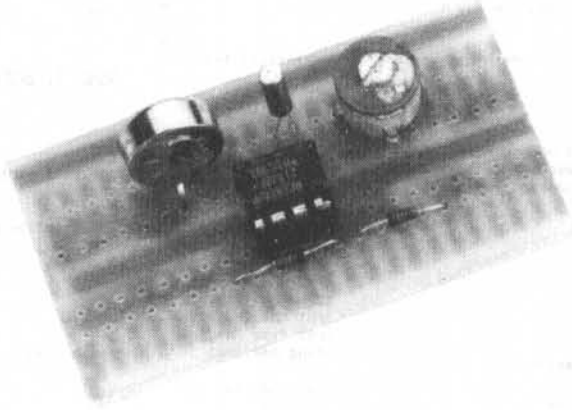


Wielu Czytelników może mieć wrażenie, że mowa będzie o jeszcze jednym czasomierzu. Tymczasem rzecz dotyczy bardzo specyficznego układu: jest to zegar skonstruowany specjalnie z myślą o współpracy z systemem mikroprocesorowym.

Zegar czasu rzeczywistego



Wbrew pozorom takie układy nie są wcale tak egzotyczne, jak mogłoby się na pierwszy rzut oka wydawać. Wystarczy przypomnieć magnetowidy z funkcją timera czy PC-ta. W urządzeniach tych z reguły znajduje się specjalna kostka - zegar czasu rzeczywistego. Wiele firm na świecie ma w swojej ofercie takie układy scalone - nam jako przykład posłuży wyrób firmy PHILIPS oznaczony symbolem PCF8583.

Układ jest szeregowo programowany za pośrednictwem szyny I²C, co sprawia, że mieści się w zaledwie ośmionóżkowej obudowie. Zawiera w sobie zegar, interfejs szyny wraz z układami logiki sterującej i pamięć RAM 256 bajtów, co jest rozwiązaniem bardzo wygodnym, gdyż umożliwia łatwe zapamiętywanie nowych parametrów zegara lub innych dodatkowych informacji. Oprócz końcówek zasilania, kostka ma wyprowadzenia dla dołączenia kwarcu lub zewnętrznych impulsów taktujących, przełączany adres A0, szynę I²C oraz wyjście przerwania INT. Układ może działać nie tylko jako zegar z alarmem, może także odmierzать zaprogramowane odcinki czasu (funkcja timera) lub zliczać impulsy doprowadzone do wejścia OSC1. Pracą układu steruje szesnaście rejestrów, które umieszczone są w bloku pamięci RAM począwszy od adresu 00H. Wszystkie rejestry mogą być odczytywane i zapisywane. Przyjęto założenie, że rejestry mają postać bajtowa, a funkcjami układu sterują poszczególne bity w re-

jestrze. Cyfry zapisuje się w postaci BCD. W przypadku rejestru daty oznacza to, że cztery najmłodsze bity rejestru będą oznaczały liczbę jednostek, a cztery starsze liczbę dziesiątek dni ustawianej daty. Poniżej przedstawiono wykaz rejestrów wraz z ich adresami oraz opisem bitów sterujących funkcjami układu.

00H - rejestr kontrolny:

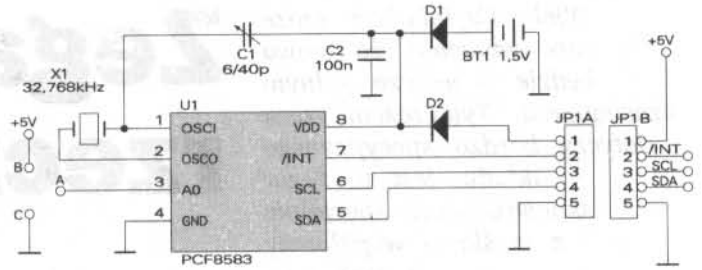
- b0: flaga alarmu timera,
- b1: flaga alarmu zegara,
- b2: zezwolenie na alarm. Jeżeli bit jest wyzerowany, b0 zmienia cyklicznie swój stan logiczny co 0,5s, a b1 co 0,5min.,
- b3: przy odczycie maskowanie bitów roku i tygodnia,
- b4, b5: sterowanie oscylatorem: 00 - dołączony kwarc, 01 - do wejścia OSC1 doprowadzony sygnał 50Hz, 10 - wejście OSC1 zlicza impulsy,
- b6: gdy bit jest ustawiony, zegar nie zlicza,
- b7: flaga zatrzymania zliczania.

Rejestry zegara:

- 01H - rejestr ułamków sekundy:
 - b0...b3: setne części sekundy,
 - b4...b7: dziesiąte części sekundy,
- 02H - rejestr sekund:
 - b0...b3: sekundy,
 - b4...b7: dziesiątki sekund,
- 03H - rejestr minut (w konwencji jak wyżej)
- 04H - rejestr godzin:
 - b0...b3: godziny,
 - b4...b5: dziesiątki godzin,
 - b6: 0-przed południem, 1-po południu,
 - b7: 0-format 24 godzinny, 1-format

- 12 godzinny (b6 jest wtedy uwzględniany),
- 05H - rejestr daty:
 - b0...b3: dni miesiąca,
 - b4...b5: dziesiątki dni miesiąca,
 - b6, b7: numer roku (00- rok przestępny),
- 06H - rejestr miesiąca:
 - b0...b3: jednostki cyfry miesiąca,
 - b4: dziesiątki cyfry miesiąca,
 - b5...b7: dzień tygodnia (liczba od 0 do 6 zapisana binarnie), 07H - rejestr timera,
- 08H - rejestr alarmu:
 - b0...b2: bity sterujące funkcją timera (000 - timer wyłączony, 001 - timer zlicza ułamki sekundy od 0 do 99, timer zlicza sekundy itd.),
 - b3: 1 - w przypadku przepełnienia timera zostanie ustawiona flaga alarmu w rejestrze kontrolnym oraz na wyjściu INT pojawi się stan niski; 0 - zostanie ustawiona tylko flaga alarmu,
 - b4, b5: funkcje alarmu zegara: 00 - zegar pracuje bez alarmu; 01 - alarm codzienny, układ nie uwzględnia rejestrów alarmowych daty i dnia tygodnia; 10 - alarm w określonym dniu tygodnia, układ nie uwzględnia rejestru alarmowego daty, natomiast uwzględnia rejestr dnia tygodnia; 11 - alarm zgodnie z wszystkimi ustawionymi rejestrami alarmowymi (oprócz rejestru dnia tygodnia).
 - b6: włączenie alarmu timera,
 - b7: 1- w przypadku alarmu zegara ustawiony zostanie bit w rejestrze

- * procesor 8031/51 z zegarem 11MHz lub mniej
- * bajty i bity ustawiane przed wywołaniem procedury I2C
- * 2ch adres układu do zapisu/odczytu przez procedurę
- * 2bh subadres rejestru układu od którego zacznie się zapis/odczyt
- * r1 adres bufora danych do wysłania/odbioru w pamięci RAM procesora
- * r2 długość bufora danych w pamięci RAM procesora
- * 50h bit gdy 0 procedura zapisze dane z bufora do zaadresowanego układu gdy 1 procedura odczyta dane z zaadresowanego układu do bufora
- * bajty i bity używane przez procedurę
- * 52h bit flaga zakończenia transmisji gdy 1
- * 51h bit flaga nadania subadresu gdy 1
- * 2dh bajt petla 10ms
- * 2eh licznik bitów



Rys. 1. Schemat elektryczny zegara czasu rzeczywistego

```

i2c jnb 50h,i2cw
jmp i2cr

i2cw clr 52h ;zapis do pamieci eeprom
clr 51h ;wsk.zakonczenia transmisji
call i2cw1 ;wsk.nadania subadresu
ret

i2cw1 call i2ctest
jc blad
iw11 mov a,2ch
clr c
rlc a
mov 2eh,#9
clr p1.7
call i2ctakt
iw3 clr p1.6 ;SCL 1
call i2ctakt
rlc a
djnz 2eh,iw2
jmp iw21

iw21 mov p1.7,c
call i2ctakt
setb p1.6 ;SCL h
call i2ctakt
jmp iw3

iw21 setb p1.7 ;SDA h
call i2ctakt
setb p1.6 ;SCL h
jnb p1.7,iw22 ;ACK?
mov 2dh,#3

iw23a mov 2eh,#ffh
iw23 djnz 2eh,iw23a
jb p1.7,blad
iw22 jnb 51h,iw24 ;czy nadany subadres?
mov a,@r1 ;kolejny znak z bufora do A
mov 2eh,#9
iw27 jnb p1.6,iw27
jb 52h,iw28
inc r1
djnz r2,iw3
setb 52h
jmp iw3

iw28 clr p1.6 ;SCL 1
call i2ctakt
clr p1.7 ;SDA 1
call i2ctakt
setb p1.6 ;SCL h
call i2ctakt
clr a
setb p1.7 ;END
ret

iw24 mov a,2bh
setb 51h
mov 2eh,#9
iw26 jnb p1.6,iw26
jmp iw3

blad orl p1,#c0h ;blad zapisu/odczytu ukkladu
mov a,#ffh ;procedura wraca
ret ;z wartoscia ffh w A

i2ctakt ;opoznienie
nop
nop
ret

i2ctest
mov a,p1
anl a,#c0h
cjne a,#c0h,i2ct1 ;SCL i SDA wolne?
i2ct5 clr c ;SCL i SDA wolne
ret

i2ct1 mov 2dh,#3
i2ct3 mov 2eh,#ffh
i2ct2 djnz 2eh,i2ct2
djnz 2dh,i2ct3
mov a,p1
anl a,#c0h
cjne a,#c0h,i2ct4
jmp i2ct5
i2ct4 setb c ;blad SCL i SDA nadal zajete

ret
    
```

Rys. 2. Procedura assemblerowa zapisu i odczytu danych przez magistralę I2C

kontrolnym i podany stan niski na wyjście INT, 0- ustawiony zostanie tylko bit w rejestrze kontrolnym.

Rejestry alarmu:

09H - rejestr alarmowy ułamków sekund

0AH - rejestr alarmowy sekund

0bH - rejestr alarmowy minut

0CH - rejestr alarmowy godzin

0DH - rejestr alarmowy dni (bez numeru roku)

0EH - rejestr alarmowy miesiąca/dnia tygodnia. W zależności od ustawionej opcji alarmu zawiera albo numer miesiąca, albo numer dnia tygodnia. W drugim przypadku funkcje bitów rejestru są następujące:

b0: alarm, jeśli dzień tygodnia 0,

b1: alarm jeśli dzień tygodnia 1 itd.

b7: nie używany

0FH - rejestr alarmowy timera.

Struktura rejestrów alarmu jest identyczna z rejestrami zegara. Służą do porównania z rejestrami zegara w przypadku włączonej funkcji alarmu. Zliczanie czasu przez zegar nie powoduje zmiany ich zawartości.

W związku z opisanymi bitami jest niezbędna mała uwaga. Układ alarmu zegara może pracować w czterech trybach:

1. alarm wyłączony - zegar tylko odmierza czas,

2. alarm jednorazowy - układ komparatora porównuje rejestry zegara począwszy od rejestru ułamków sekund aż po rejestr miesiąca wraz z odpowiadającymi im rejestrami alarmu, które zostaną opisane poniżej. W przypadku stwierdzenia równości wszystkich rejestrów zostaje ustawiona flaga alarmu i ewentualnie podany stan niski na wyjście INT.

3. alarm codzienny - z porównania wykluczone zostaną rejestry daty, a alarm zostanie włączony, jeśli będą zgodne ustawienia rejestrów godzin, minut itd.

4. alarm cotygodniowy - układ pomija rejestry daty, natomiast uwzględnia przy porównaniu rejestr zawierający numer dnia tygodnia, co jest bardzo wygodnym rozwiązaniem, jeśli układ ma działać cyklicznie; staje się zbędne wyliczanie kolejnej daty alarmu.

Pozostałe 240 bajtów pamięci RAM, począwszy od adresu 10H, może być dowolnie wykorzystane. Sekwencja programowania alarmu powinna przebiegać w ten sposób, że najpierw zapisuje się wybrane parametry czasu i daty alarmu. Jeżeli będzie to alarm cykliczny, cotygodniowy, pod adresem 0EH trzeba ustawić jeden bit odpowiadający wybranemu dniu tygodnia. W rejestrze 08H bity dotyczące timera powinny być wyzerowane, natomiast ustawienie pozostałych określi rodzaj alarmu oraz zezwolenie na pojawienie się stanu niskiego na wyjściu INT. Na koniec w rejestrze 00H trzeba wyzerować bity 0 i 1. W przypadku alarmu są one ustawiane i można wyzerować je powtórnie tylko programowo.

Na **rysunku 1** przedstawiono schemat elektryczny układu. Ogniwo 1,5V służy do podtrzymania działania układu i pamięci RAM, jeżeli zasilanie +5V musi być na pewien czas wyłączone. Przy zasilaniu obniżonym napięciem (minimum 1V) prąd pobierany przez układ zawiera się w granicach 2...10µA, dlatego ogniwo podtrzymujące wystarcza na długo. Jeżeli nie zależy nam na podtrzymywaniu pracy zegara przy zanikach zasilania, można w układzie pominąć ogniwo i diody, a +5V podłączyć bezpośrednio do nóżki 8 układu. Przełączana zwora ABC służy do ustalenia adresu w obrębie sieci I²C. Zwykle zwora powinna łączyć punkty A i C. Jeżeli do sieci dołączony jest inny układ o adresie A0H (np. drugi zegar, pamięć EEPROM itd.), to zwora powinna połączyć punkty A i B - adres wywołania układu będzie wtedy ustawiony na A2H. Wyjście INT jest typu „otwarty dren“ i w zależności od typu współpracujących z modułem zegara układów trzeba je połączyć z napię-

WYKAZ ELEMENTÓW

Kondensatory

C1: 6/40pF

C2: 100nF

Półprzewodniki

U1: PCF8583 (Phillips)

D1, D2: 1N4001

Różne

X1: rezonator kwarcowy 32,768kHz

BT1: bateria 1,5V

JP1A, JP1B: złącze x5

ciem +5V poprzez rezystor o wartości 1...10kΩ. Układ składa się zaledwie z kilku elementów, można go więc z łatwością zmontować na kawałeczku płytki uniwersalnej.

Po zmontowaniu układu i dołączeniu napięcia zasilającego podstawową oznaką świadczącą, że układ działa, jest pojawienie się na wyjściu INT impulsów o częstotliwości 1 Hz i wypełnieniu 50%. Jest to stan RESET dla układu. Rejestry wewnętrzne można odczytywać i zapisywać przy napięciu zasilającym +5V.

Układ zegara czasu rzeczywistego jest urządzeniem przydatnym w sterownikach wszelkiego typu. Pozwala procesorowi precyzyjnie „śledzić czas“ i programować sekwencje alarmowe nawet na rok w przód. Autor dołączył opisany moduł zegara do układu tunera satelitarnego, opisanego w EP 9/93, wzbogacając go w ten sposób o funkcję programowanego timera umożliwiającą automatyczne nagrywanie na magnetowidzie programów satelitarnych.

Jako uzupełnienie artykułu zamieszczamy wydruk procedury assemblerowej (**rys. 2**), która przez magistralę I²C umożliwia procesorowi zapis i odczyt danych z dowolnego układu podporządkowanego. Port PI.6 pełni funkcję końcówki SCL, a PI.7 końcówki SDA. Po asemblacji procedura powinna mieć długość 327 bajtów.

Ryszard Szamaniak, AVT