



Dział „Projekty Czytelników” zawiera opisy projektów nadesłanych do redakcji EP przez Czytelników. Redakcja nie bierze odpowiedzialności za prawidłowe działanie opisywanych układów, gdyż nie testujemy ich laboratoryjnie, chociaż sprawdzamy poprawność konstrukcji.

Prosimy o nadsyłanie własnych projektów z modelami (do zwrotu). Do artykułu należy dołączyć podpisane oświadczenie, że artykuł jest własnym opracowaniem autora i nie był dotychczas nigdzie publikowany. Honorarium za publikację w tym dziale wynosi 250,- zł (brutto) za 1 stronę w EP. Przesyłanych tekstów nie zwracamy. Redakcja zastrzega sobie prawo do dokonywania skrótów.

# Zegar imprezowicza

## Sterownik z kalendarzem i termometrem

*Stara, poczciwa '51-ka nie odeszła zupełnie w zapomnienie.*

*Nadal sporo tych układów zalega w szafkach konstruktorów i szkoda by było, żeby przekwalifikować je na elektroniczny złom. Może w ramach porządków oczyścimy nieco nasze magazynowe szafki, zyskując przy okazji przydatny na co dzień zegar stołowy.*

### Rekomendacje:

*projektów zegarów jest mnóstwo, ale ten wyróżnia się bogactwem funkcji, w tym również przypomina o imieninach by nie przegapić żadnej imprezki.*



Opisywane urządzenie powstało w wyniku prac prowadzonych przez autora nad innymi projektami: zegarem z dziesięcioma timerami i zegarem z komunikatami. W efekcie powstało urządzenie posiadające funkcje: zegara, kalendarza, timera (wykorzystywanego do sterowania urządzeniami elektrycznymi za pomocą podczerwieni), wyświetlania komunikatów (można wprowadzić szesnaście różnych komunikatów o długości do 24 znaków, dla każdego ustawiana jest data wyświetlenia lub częstotliwość powtarzania), budzika, termometru, termostatu (który może sterować jednym z pięciu urządzeń elektrycznych).

### Opis budowy

Schemat elektryczny zegara pokazano na rys. 1. Sercem układu jest mikrokontroler AT89C52 firmy Atmel. Współpracuje on z układami peryferyjnymi takimi jak:

- Zegar czasu rzeczywistego (PCF8583), którego pierwszych 9 bajtów pamięci jest wykorzystanych dla funkcji zegara i kalendarza. Kolejne 8 bajtów może być zaprogramowane jako rejestr alarmu lub wykorzystane jako zwykła pamięć RAM, tak jak pozostałe 240 bajtów. Układ RTC może być zasilany z baterii 3 V zapewniającej ciągłość pracy i zachowanie danych w czasie przerwy zasilania.
- Wyświetlaczalfanumeryczny(SANYODM157A) LCD 3x24 znaków, pracujący z 4-bitową szyną danych. Jego podświetlenie jest załączone na stałe, a intensywność jest regulowana

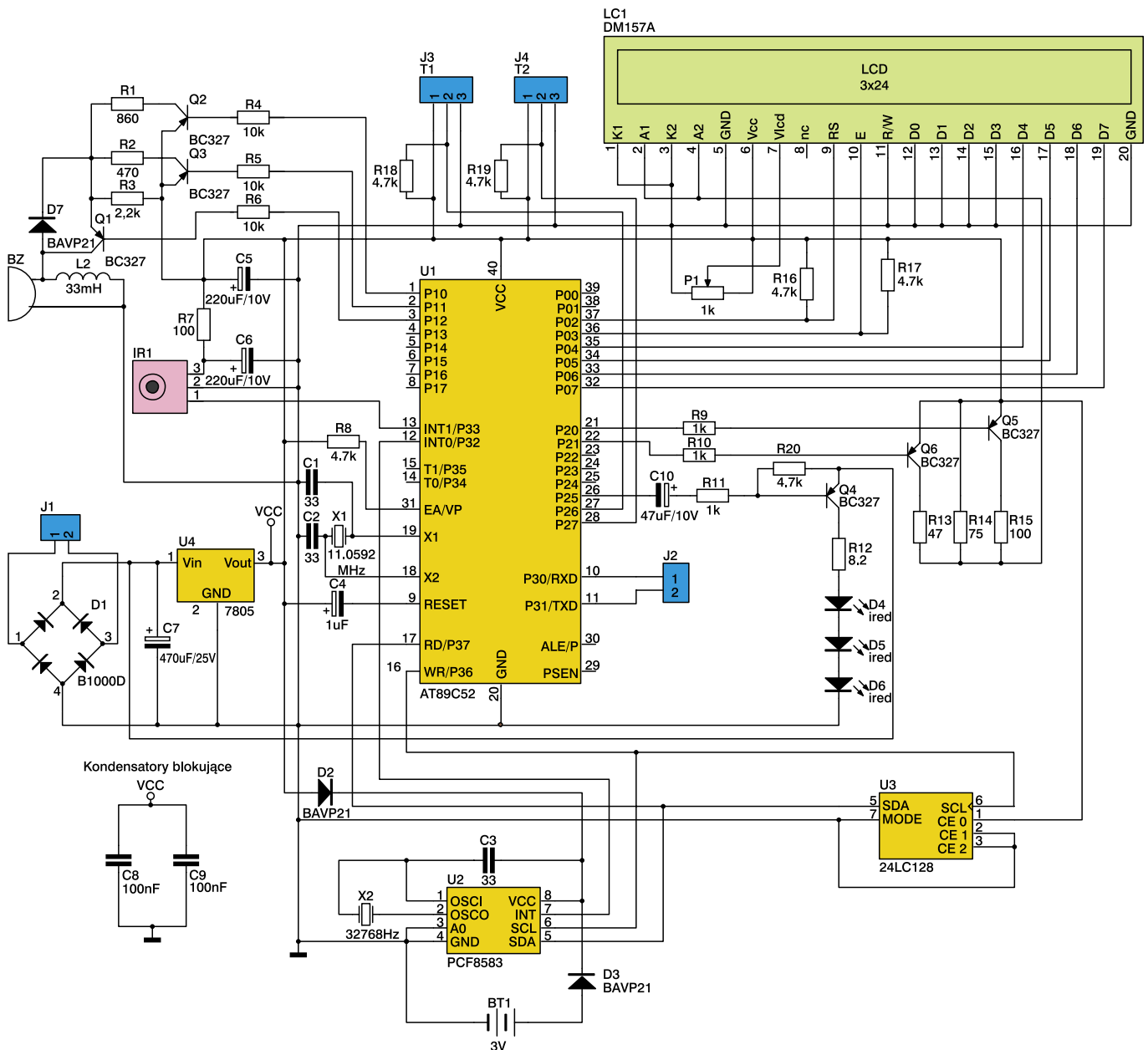
układem, w skład którego wchodzi tranzystory Q5 i Q6 oraz rezystory R13, R14 i R15.

- Odbiornik podczerwieni podłączony bezpośrednio do wejścia przerwania INT1 mikrokontrolera. Zastosowany odbiornik musi pracować w paśmie od 36 do 40 kHz, w którym pracuje większość pilotów. W celu zmniejszenia zakłóceń od układów cyfrowych zasilanie odbiornika jest filtrowane elementami R7 i C6.
- Diody nadawcze sterowane za pomocą stopnia mocy na tranzystorze Q4. Maksymalny prąd jest zależny od rezystora R12. W bazie tranzystora umieszczono kondensator, dzięki któremu diody nie ulegną uszkodzeniu nawet w przypadku pojawienia się przez dłuższy czas na wyjściu mikrokontrolera wysokiego poziomu. W celu uzyskania dużego zasięgu maksymalny prąd płynący przez diody wynosi ok. 400 mA.
- Pamięć EEPROM o pojemności 16 kB. Tak duża pamięć jest konieczna, gdyż jest w niej zapisana tabela ponad 700 imion wyświetlanych po dwa dla każdego dnia roku, nastawy timerów, komunikaty i kody pilotów dla poszczególnych urządzeń. Pamięć przed zamontowaniem musi być wstępnie zaprogramowana plikiem zawierającym zapisane imiona. Pozostałe dane są zapisywane przez mikrokontroler w różnych opcjach menu.
- Buzzer do sygnalizacji dźwiękowej. Na tranzystorach Q2 i Q3 oraz rezystorach R1...R3 jest realizowana regulacja głośności. Tranzystor Q1 wzmacnia sygnał generowany przez mikrokontroler.

## Projekt 169

### PODSTAWOWE PARAMETRY

- Płytką o wymiarach 142x60 mm
- Zasilanie zalecane 12 VDC, może być również 8,5 VAC
- Pamięć nieulotna na bazę imion, notatki, kody pilotów: EEPROM 16 kB
- Sygnalizacja dźwiękowa
- Sterowanie zegarem za pomocą pilota RC5
- Możliwość wyświetlania komunikatów (notatek) użytkownika przypisanych do daty
- Sterowanie urządzeniami elektrycznymi (np. telewizorami) poprzez nadajnik podczerwieni, zegar w tym trybie symuluje działanie pilota RC5
- Możliwość definiowania kodów różnych typów pilotów RC5
- Funkcja alarmu
- Funkcja termostatu (sterowanie grzałką poprzez tor IR)



Rys. 1. Schemat elektryczny

- Czujniki temperatury. W zależności od wersji oprogramowania mogą to być układy DS18B20 lub DS18B20. Są one dołączone do złączy J3 i J4 za pomocą linii trójżyłowej, co pozwala umieścić czujniki w dużej odległości od zegara. Podstawowa wersja oprogramowania obsługuje tylko jeden czujnik DS18B20.
- Zasilacz składa się z mostka prostowniczego i stabilizatora 5 V oraz kondensatorów filtrujących. Urządzenie można zasilać bezpośrednio z transformatora dającego na wyjściu napięcie ok. 8,5 V. Ze względu na nadajniki podczerwieni zalecane jest jednak stosowanie zewnętrznego zasilacza stabilizowanego 12 V. Można wtedy zrezygnować z mostka prostowniczego.
- Klawiatura – jej funkcję pełni pilot RC5. W urządzeniu modelowym jest to pilot do telewizorów firmy Philips, popularnie zwany „bananem”.

- Pamięć dodatkowa. Oprócz pamięci zawartej wewnątrz mikrokontrolera mamy do dyspozycji jeszcze dwie dostępne poprzez szynę I<sup>2</sup>C. Są to: pamięć RAM w układzie PCF8583 wykorzystywana jako podręczny schowek dla bufora oraz pamięć EEPROM w układzie 24C128.

### Program

Wszystkie listingi programu zostały zamieszczone na płytce CD-EP12/2008. Z uwagi na ograniczone miejsce nie zamieszczamy ich w artykule.

Po włączeniu zasilania program ustawia początkowe wartości dla niektórych zmiennych i wskaźników, konfiguruje timery mikrokontrolera, a następnie inicjalizuje wyświetlacz oraz zegar czasu rzeczywistego. Podczas dalszej pracy wykonywana jest nieskończona pętla `while`, wewnątrz której jest zagnieżdżona kolejna,

której wykonanie jest uzależnione od ustawienia wskaźnika `petla`. Wskaźnik ten może być ustawiony podczas obsługi przerwania `INT0` zgłaszanego przez zegar RTC, po odebraniu rozkazu w kodzie `RC5` lub w niektórych funkcjach menu. Ustawienie wskaźnika powoduje wykonanie instrukcji `switch` sterującej wykonywaniem programu w zależności od zmiennej `status_g`. Po wykonaniu instrukcji, przed wyjściem z pętli, wskaźnik `petla` zmienia wartość na przeciwną. Ma to na celu zablokowanie kolejnego wykonania tej pętli lub, jeśli w trakcie jej wykonywania jego wartość została wyzerowana, umożliwienie powrotu do niej. Jeżeli razem z wyzerowaniem tego wskaźnika nastąpiło przypisanie zmiennej `status_g` innej wartości, przy następnym wykonaniu pętli zostanie zrealizowany inny element instrukcji `switch`. Taka konstrukcja umożliwia skoki programu do obsługi wyświetlania czasu, wyświetlanie i poruszanie się po różnych



Rys. 2. Opis funkcji przycisków pilota

poziomach menu. Wartość zmiennej `status_g` decyduje o funkcjach wykonywanych przez program i jest to główny wskaźnik sterujący jego pracą.

Podstawowym trybem pracy jest oczekiwanie na przerwanie INTO od zegara RTC. W obsłudze przerwania jest ustawiany `status_g=1` oraz wskaźnik `petla=1`, czyli zostają spełnione warunki wejścia do pętli. Teraz za pośrednictwem szyny I<sup>2</sup>C następuje odczyt rejestrów zegara. W jednym rejestrze układu PCF8583 jest przechowywanych kilka danych, dlatego trzeba je rozdzielić i umieścić w odpowiednich zmiennych, za co odpowiada funkcja `CzytajCzas()`. Oddzielnie jest wyliczany i przechowywany rok, a to dlatego, iż w tym układzie jest on zliczany tylko do czterech, co wystarcza jedynie do wyliczenia roku przestępnego. Teraz program przechodzi do wyświetlenia na wyświetlaczu aktualnego czasu (`LcdWyświetl()`). Funkcja ta odpowiada za wyświetlenie wszystkich danych, jednak nie jest to możliwe jednocześnie. O tym, które dane zostaną wyświetlone decydują wskaźniki. Kolejną czynnością jest wyliczenie pozycji komunikatu na wyświetlaczu. Wyświetlanie komunikatu rozpoczyna się o godzinie 24.00 i jest kontynuowane przez cały dzień, o ile wcześniej nie zostanie on skasowany przez użytkownika. Treść komunikatu może mieć do 24 znaków. Jednego dnia może być wyświetlonych nawet 16 komunikatów, tzn. tyle, ile można wprowadzić do pamięci. Komunikaty są kolejno przesuwane na dolnej linijce wyświetlacza od prawej do lewej o jeden znak, co sekundę.

Funkcja `SprawdzNapis()` odczytuje kolejno treść komunikatów (wraz z danymi o terminie wyświetlania) z pamięci EEPROM. Po sprawdzeniu zgodności daty komunikat jest ładowany do bufora. Dla płynnego wyświetlania kolejnych napisów potrzebny jest bufor o długości dwukrot-

nie większej od liczby znaków w jednej linijce wyświetlacza. Pierwszy komunikat jest ładowany do pierwszej połowy bufora, następny do drugiej. Podczas wyświetlania, z bufora są pobierane zawsze 24 znaki (długość linijki wyświetlacza). Wskaźnik początku wybierania jest inkrementowany do 47. W chwili osiągnięcia przez niego połowy długości bufora na wyświetlaczu jest wyświetlany w całości drugi komunikat. W tym momencie jest poszukiwany kolejny komunikat spełniający kryteria daty, który następnie zostanie załadowany w buforze tym razem do pierwszej połowy. Cykl ten powtarza się przez cały czas wyświetlania komunikatów. Szczególnym, choć najczęściej występującym zdarzeniem, jest wyświetlanie tylko jednego komunikatu. W takim przypadku dla uzyskania efektu płynnego przewijania jest on ładowany zarówno do pierwszej, jak i do drugiej połowy bufora.

Funkcja `SprawdzTimer()` jest wywoływana co minutę. Ona też na początku wylicza adres w pamięci EEPROM, z którego są odczytywane nastawy timerów. Teraz następuje sprawdzenie odczytanych nastaw z aktualnym dniem tygodnia i czasem. Najpierw są sprawdzane warunki załączenia, a następnie wyłączenia timera. Jeśli są warunki zgodne, rozpoczyna się wykonywanie kolejnych instrukcji powodujących w efekcie wyświetlenie lub wygaszenie ikonki dzwonka na wyświetlaczu i wysłanie za pomocą podczerwieni kodu do odpowiedniego urządzenia. Ciekawostką są tutaj dwie funkcje `ZapiszBuforRAM()` i `CzytajBuforRAM()`. Do przechowywania treści komunikatu i kodu pilota odczytanego z pamięci EEPROM jest wykorzystywany ten sam bufor. W momencie nadawania kodu pilota zachodzi tu konflikt. Dlatego przed nadaniem kodu, fragment bufora, w którym są zapisane komunikaty jest przepisywany do pamięci RAM układu PCF8583, a po jego nadaniu jest odtwa-

rzana poprzednia zawartość bufora. Dzięki temu na wyświetlaczu nie ma żadnych zakłóceń. Nadaniu kodu towarzyszy dodatkowo krótki sygnał dźwiękowy. Jeśli kilka timerów jest ustawionych na tę samą godzinę, ich kody będą wysyłane w takiej kolejności, w jakiej są odczytywane z pamięci.

Funkcja `SprawdzAlarm()` ma podobną budowę, choć jest znacznie prostsza. Jej działanie ogranicza się tylko do sprawdzenia zgodności odczytanych nastaw z aktualnym czasem i ustawienia znacznika załączającego sygnał dźwiękowy.

Funkcja `Temperatura()` odczytuje zawartość rejestrów czujnika DS18B20 i umieszcza je w buforze poczynając od adresu 50 (za obszarem, w którym są przechowywane komunikaty). Teraz następuje sprawdzenie znaku odczytanej temperatury i odpowiednie ustawienie wskaźnika minus. Kolejne czynności to wyliczenie dziesiątych części stopnia i przekształcenie temperatury do postaci dziesiętnej. Na koniec zostaje zainicjalizowany kolejny pomiar temperatury. Czynność ta jest wykonywana na końcu, ponieważ czujnik potrzebuje ok. 750 ms na pomiar temperatury. Odczyt temperatury następuje co osiem sekund. Tak skonstruowana funkcja wyświetla więc wartość odczytaną osiem sekund wcześniej, jednak w przypadku temperatury nie ma to istotnego znaczenia.

Po odczytaniu temperatury program przechodzi do funkcji `SprTermo()`, w której sprawdza czy jest ustawiony znacznik termostatu. Jeśli tak, następuje porównanie aktualnej temperatury z temperaturą załączenia i wyłączenia termostatu. Niezależnie od ustawienia znacznika termostatu, w funkcji tej następuje też porównanie odczytanej temperatury z maksymalną i minimalną, jakie zostały zanotowane poprzednio, i wartości te są ewentualnie modyfikowane. Efektem działania tej funkcji, w przypadku stwierdzenia przekroczenia ustawionej temperatury, jest wysłanie odpowiedniego kodu do sterowanego urządzenia. Temperatura jest wyświetlana tylko w przypadku, gdy jest podłączony czujnik. Program automatycznie rozpoznaje jego obecność i ustawia wskaźnik wyświetlania temperatury.

## Obsługa zegara

Na rys. 2 przedstawiono opis funkcji przycisków pilota zastosowanego do sterowania zegarem.

Równoczesne naciśnięcie przycisków *strojenie* na pilocie powoduje wejście do menu. Teraz przyciskami + i – możemy przeglądać pierwszy poziom menu. Z tego poziomu można przejść do ustawiania komunikatów, timerów, czasu i daty, termostatu, poziomu sygnału dźwiękowego, budzika i zapisu kodu pilota. Do wybranej przez nas opcji wchodzimy naciskając przycisk *menu* na pilocie.

Menu ustawiania komunikatów umożliwia wybranie daty wyświetlania komunikatu, częstotliwości jego powtarzania i jego treści. Treść

komunikatu jest wprowadzana za pomocą klawiatury numerycznej na podobnej zasadzie jak w telefonach komórkowych. Po pierwszym naciśnięciu przycisku zaczyna się odliczanie czasu. Po skończeniu odliczania znak zostanie zapisany do bufora, w którym jest przechowywana treść komunikatu, a następnie kursor zostanie inkrementowany. Jeśli przed upływem tego czasu przycisk zostanie ponownie naciśnięty, na wyświetlaczu zostanie wyświetlony kolejna litera ukryta pod tym przyciskiem. Czas oczekiwania na kolejne naciśnięcie wynosi ok. 1 sekundy. W przypadku błędnego wprowadzenia znaku możemy go wykasować przyciskiem 0. Po wprowadzeniu kompletnego komunikatu i sprawdzeniu terminu zatwierdzamy ustawienia przyciskiem *menu*. Zatwierdzenie powoduje zapisanie ustawień i treści komunikatu do pamięci EEPROM.

Menu ustawienia timerów umożliwia przeglądanie i zmianę nastaw poszczególnych timerów.

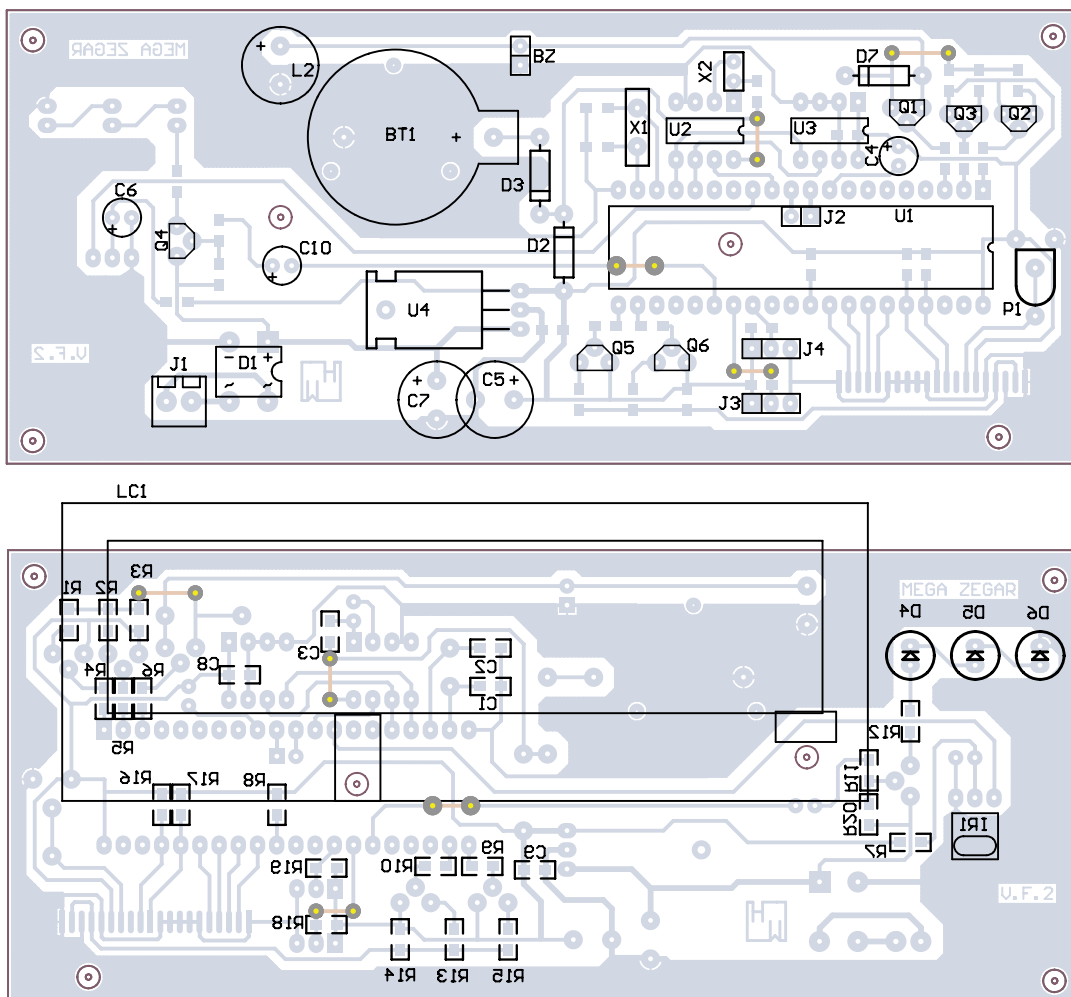
Podobną konstrukcję posiadają menu: czasu i daty, termostatu i ustawiana budzika. Bardzo prostą konstrukcję posiada menu ustawiania głośności sygnału dźwiękowego. Po jego wybraniu na wyświetlaczu pojawia się napis „poziom dźwięku”. Teraz przyciskami + i - zmieniamy stany na wyjściach P00 i P01 portu P0. Wyjścia

te sterują tranzystorami Q2 i Q3 łączącymi rezystory połączone między zasilanie a przetwornik. W taki sam sposób na tranzystorach Q5, Q6 jest realizowana regulacja jasności podświetlenia wyświetlacza. Ta regulacja jest jednak dostępna bez potrzeby wchodzenia do menu.

Ostatnim, do tego najbardziej skomplikowanym, jest menu zapisu kodu pilota. Mamy możliwość zaprogramowania kodu załączenia i wyłączenia pięciu dowolnych urządzeń sterowanych podczerwienią, co daje w sumie dziesięć różnych kodów. Oprócz zapisu kodu można także w tym menu przetestować prawidłowość wcześniej zapisanego kodu wymuszając jego nadanie. Po wybraniu tego menu jesteśmy na poziomie umożliwiającym wybór urządzenia, którego kod chcemy zapisać lub przetestować. Nazwy są stałe (TV, TU, MA, R1, R1), co oczywiście nie zabrania np. pod nazwą TU zaprogramować kod dla telewizora. Po wybraniu urządzenia naciskając przycisk *menu* na pilocie przechodzimy na następny poziom, na którym mamy trzy możliwości wyboru. Naciskając 1 przechodzimy do zapisu kodu pilota, 2 możemy sprawdzić kod zapisany pod tą nazwą, przycisk *standby* umożliwia powrót na poziom wyboru urządzenia. Zapis kodu jest operacją bardzo prostą. Po wybraniu 1, na wyświetlaczu pojawia się napis „CZEKAJ”, wyświetlany przez ok. 3 sek. Jest to

czas na puszczenie przycisku pilota sterującego urządzeniem. Po tym czasie pojawia się napis „wprowadz kod dla ON”. Teraz pilot, którego kod chcemy zapisać kierujemy w stronę zegara i naciskamy krótko przycisk włączający sprzęt. Po sekundzie pojawi się napis „wprowadz kod dla OFF”. Należy teraz nacisnąć odpowiedni przycisk i cała operacja jest już zakończona. Zostaje wyświetlone poprzednie menu. Warto jest sprawdzić, czy kod został prawidłowo zapisany i czy zegar jest w stanie sterować np. naszym telewizorem. Pilotem od zegara wybieramy 2 (test kodu), pojawia się napis „kod dla ON”, aby go nadać naciskamy przycisk *menu*, wtedy pojawia się napis „CZEKAJ” i po ok. 1,5 sek. zostanie wysłany zaprogramowany kod. Urządzenie powinno się włączyć, a na wyświetlaczu pojawia się napis „kod dla OFF”. Znowu naciskamy przycisk menu, napis „CZEKAJ” i urządzenie powinno się wyłączyć. Wychodzimy z tej opcji przyciskiem *standby*.

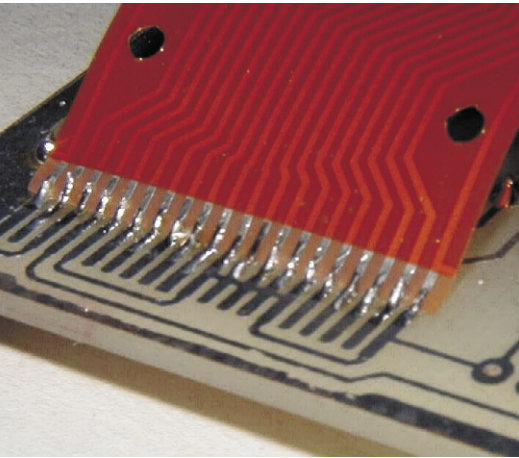
Ze względu na dużą różnorodność kodów nadawanych przez piloty nie ma uniwersalnych procedur, które umożliwiłyby ich odczyt. Jednak w tym przypadku nie ma potrzeby „rozumienia” zapisywanego kodu, wystarczy tylko zapisać jego przebieg w taki sposób, aby można go było później odtworzyć. Do tego celu wystarczy próbować sygnał z odbiornika podczerwieni ze stałą częstotliwością. Wynik próbkowania należy zapisać najpierw w pamięci RAM, a następnie przesłać go do pamięci EEPROM. Pozostało już tylko dobrać częstotliwość próbkowania i opracować sposób efektywnego zapisu informacji uzyskanych podczas próbkowania do pamięci. Czas nadawania kompletnego kodu w zależności od systemu waha się od kilkunastu do ponad stu milisekund, natomiast szerokość pojedynczego impulsu może mieć tylko 100  $\mu$ s. Dla zachowania wystarczającej dokładności przy odtwarzaniu przebiegu konieczne jest kilkukrotne próbkowanie nawet w największych impulsach. Niewielka pamięć RAM, jaką dysponują mikrokontrolery serii 8052 bardzo utrudnia zarezerwowanie odpowiedniego bufora. Do zapisu udało mi się wygospodarować tylko 150 bajtów. Pojawił się problem, w jaki sposób zakodować jak największą ilość informacji w tym buforze. Ostatecznie przyjąłem zasadę, iż w nieparzystych bajtach



Rys. 3. Rozmieszczenie elementów na płycie drukowanej



Na CD karty katalogowe i noty aplikacyjne elementów oznaczonych na Wykazie Elementów kolorem czerwonym



Fot. 4. Sposób wlotowania taśmy wyświetlacza

**WYKAZ ELEMENTÓW**

**Rezystory**

- R1: 860 Ω
- R2: 470 Ω
- R3: 2,2 kΩ
- R4...R6: 10 kΩ
- R7: 100 Ω
- R8: 4,7 kΩ
- R9...R11: 1 kΩ
- R12: 10 Ω
- R13: 47 Ω
- R14: 75 Ω
- R15: 100 Ω
- R16...R20: 4,7 kΩ
- P1: 1 kΩ

**Kondensatory**

- C1...C3: 33 pF
- C4: 1 μF/16 V
- C5, C6: 220 μF/16 V

C7: 1000 μF/16 V

C8, C9: 100 nF

C10: 47 μF/16 V

**Półprzewodniki**

- D1: mostek B1000D
- D2, D2, D3, D7: BAVP21
- D4...D6: dioda podczerwona
- T1...T6: BC327
- U1: AT89C52
- U2: PCF8583
- U3: 24LC128
- U4: 7805
- IR1: Odbiornik podczerwieni
- Inne**
- BT1: Podstawka na baterię CR2025
- J1: ARK2
- J3, J4: Gold pin
- L2: 33 mH
- LC1: DM157A

będzie zapisana informacja o czasie trwania niskiego poziomu, a w parzystych o czasie wysokiego. Taki sposób zapisu przy częstotliwości próbkowania ok. 76 kHz pozwolił na rejestrację ponad 100 ms przebiegu. Częstotliwość próbkowania nie jest przypadkowa. Wysyłając kod za pomocą podczerwieni musimy mieć przebieg zawierający częstotliwość nośnej zmodulowany kodem zapisanym w pamięci EEPROM. Częstotliwość nośnej dla większości pilotów zawiera się w przedziale 36...40 kHz, dlatego postanowiłem wybrać częstotliwość ze środka przedziału. Próbkowanie z częstotliwością 76 kHz umożliwi później w prosty sposób uzyskać częstotliwości nośnej 38 kHz.

Zastosowana w programie metoda zapisu i nadawania zapisanego kodu sprawdza się w większości przypadków, nie zapewnia jednak stu procentowej skuteczności z dwóch następujących powodów: do włączenia niektórych urządzeń potrzebne jest nadawanie kodu przez czas dłuższy niż 100 ms, (co ze względu na pojemność pamięci jest niemożliwe); niektóre kody posiadają bity zmieniające swoją wartość przy kolejnych przyciśnięciach, czego przy tej metodzie zapisu też nie uda się uzyskać.

**Montaż i uruchomienie**

Układ został zamontowany na jednostronnej płytce drukowanej, przedstawionej na rys. 3. Znajdują się na niej wszystkie elementy wyszczególnione na schemacie za wyjątkiem buzzera.

Wskazane jest umieszczenie mikrokontrolera oraz pamięci EEPROM w podstawkach. Po zmontowaniu płytki i włożeniu do podstawek zaprogramowanych układów, jedyną regulacją, jaką należy przeprowadzić jest ustawienie kontrastu wyświetlacza. Największym problemem było połączenie wyświetlacza z resztą układu. Zdecydowałem się na dość nietypowy wyświetlacz 3x24 ze względu na jego niską cenę, a także z racji na to, że na wyświetlaczu z dwoma liniami jest za mało miejsca do wyświetlenia wszystkich danych, zaś na czterech liniijkach wyświetlacza, bez zróżnicowania wielkości liter, trudno jest jednym rzutem oka ogarnąć wszystkie informacje. Uznałem więc, że trzy linie będą optymalne do mojego zegara. Pewien problem pojawił się z podłączeniem tego wyświetlacza, co jest spowodowane dość nietypowym rozstawem pinów w zastosowanej taśmie. Trudno było dobrać do niej odpowiednie złącze. Ostatecznie na płytce drukowanej zaprojektowałem złącze z rozstawem odpowiadającym taśmie i przylutowałem ją bezpośrednio do płytki (fot. 4). Wyświetlacz jest mocowany do plastikowej ramki. Ramkę przycięłem do wymiarów samego wyświetlacza i wierząc w niej otwory zamocowałem śrubami do płytki. Dzięki takiemu zamontowaniu wyświetlacza powstał zwarty moduł łatwy do zamontowania w samodzielnej obudowie lub jako element innego urządzenia. W celu zwiększenia zasięgu działania, diody IR można zamontować na oddzielnej płytce, a tak

wykonany moduł umieścić w innym, dogodnym miejscu zapewniającym odpowiedni zasięg. Załączone oprogramowanie obsługuje tylko jeden czujnik temperatury, jednak na płytce jest przewidziane miejsce do podłączenia drugiego.

**Uwagi**

Na koniec chciałbym zwrócić uwagę na problem, jaki może wystąpić podczas uruchamiania urządzenia, polegający na braku sterowania z pilota. Jest on spowodowany procedurą odczytu kodu pilota sterującego urządzeniem. Ponieważ synchronizacja pętli próbkującej stan wyjścia odbiornika podczerwieni odbywa się tylko na zbroczu opadającym pierwszego odebranego impulsu, czas działania tej pętli musi być bardzo precyzyjnie dobrany do częstotliwości pracy pilota, jakim dysponujemy. Oryginalne piloty firmy Philips mają okres równy 1,73 ms, natomiast piloty zastępcze, które można kupić w wielu sklepach mogą mieć inną częstotliwość pracy i wtedy trzeba zmienić wartość nastaw timera w procedurze odbioru kodu.

Wąski zakres synchronizacji z pilotem ma swoje wady i zalety. Do wad oczywiście należy problem „dostrojenia” pilota i urządzenia, ale zaletą jest to, iż przełączając telewizor nie przedstawimy przy okazji zegara, jeśli piloty pracują na różnych częstotliwościach.

Henryk Więsyk  
henry1wie@op.pl

REKLAMA

# Termostat elektroniczny AVT950/1

**AVT-Korporacja Sp. z o.o.**  
03-197 Warszawa, ul. Leszczyńska 11  
tel. 022 257 84 50, fax 022 257 84 55  
e-mail: handlowy@avt.pl

**Dostępne wersje:**  
A - płytka drukowana: 36zł  
B - komplet elementów: 95zł  
C - układ zmontowany: 143zł

[www.sklep.avt.pl](http://www.sklep.avt.pl)