

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.

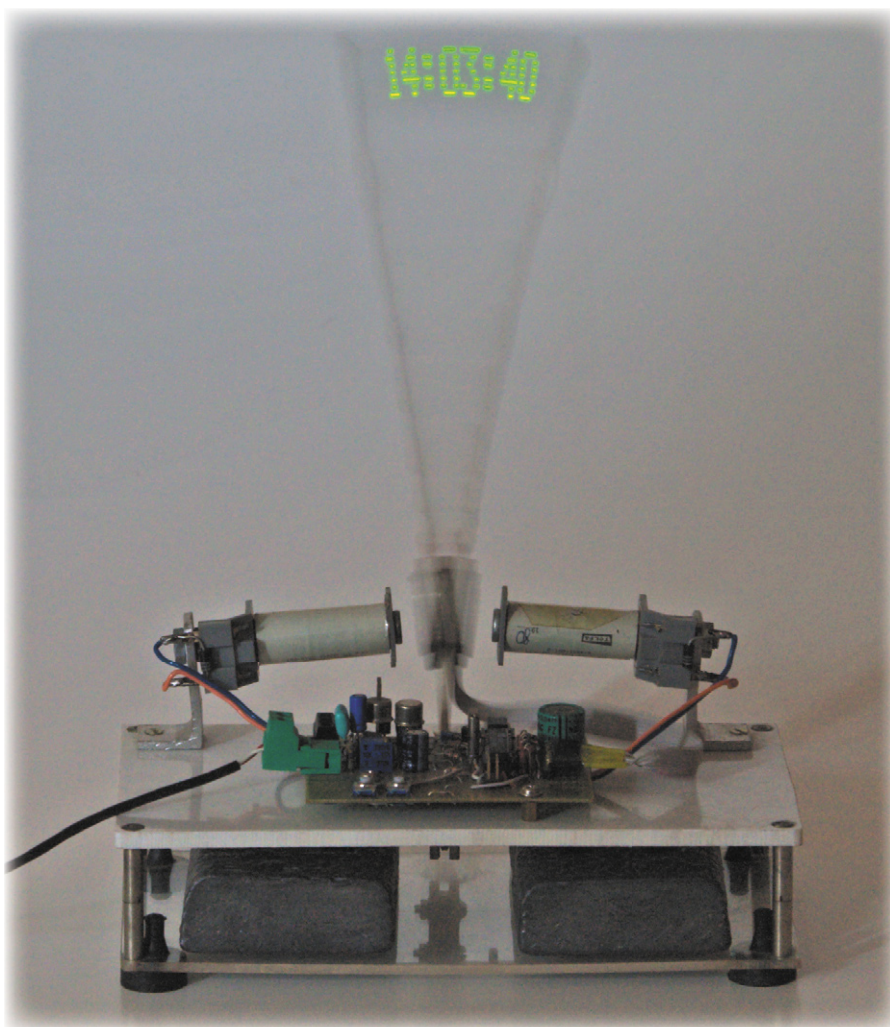
Prosimy o nadsyłanie własnych projektów z modelami (do zwrotu). Do artykułu należy dołączyć podpisane oświadczenie, że artykuł jest własnym opracowaniem autora i nie był dotychczas nigdzie publikowany. Honorarium za publikację w tym dziale wynosi 250,- zł (brutto) za 1 stronę w EP. Przesyłanych tekstów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

Wahadłowy zegar widmowy

Opisane urządzenie, jak nie trudno się domyślić, służy do wyświetlania aktualnego czasu. Jednak funkcja ta jest spełniana w bardzo nietypowy sposób, ponieważ czas w standardowym formacie HH:MM:SS jest wyświetlany za pomocą jedynie siedmiu diod LED. Poza tym, świecące cyfry wydają się być zawieszone w przestrzeni. Uzyskanie takiego efektu jest możliwe dzięki opóźnieniu ludzkiego oka w reakcji na zanik światła.

Dodatkowe
materiały na CD

Projekt
171



Linijka diod LED zamocowana jest na końcu elastycznego, plastikowego ramienia, który wprawiany jest w ruch wahadłowy za pomocą elektromagnesów (stąd „wahadłowy” w tytule). Cyfry nie są wyświetlane w całości lecz linia po linii, jednak to przemiatanie następuje tak szybko, że obserwatorowi wydaje się, że ma przed sobą cały nieruchomy obraz. Dokładnie ta sama zasada obowiązuje w telewizorze, gdzie obraz również jest wyświetlany liniami, z tym że poziomymi (w dzisiejszych czasach należałoby już wspomnieć, że chodzi o staroświecki telewizor z kineskopem ☺). Zapewne niektórym czytelnikom znane są inne zegary lub wyświetlacze widmowe oparte na tej samej zasadzie, jednak zazwyczaj są to konstrukcje gdzie linijka diod jest zamontowana na wirniku silnika elektrycznego.

To niestety wymaga albo umieszczenia całego układu na płytce z diodami i nie zawsze łatwego wyważenia go, albo zastosowania skomplikowanego systemu szczotek do przekazywania zasilania poszczególnym diodom świecącym. W tym pierwszym przypadku również potrzebne są szczotki lub transformatory obrotowe, bo układ musi przecież jakoś być zasilany. Dodatkowym problemem jest ustawianie godziny – o przyciskach na płytce można zapomnieć, bo przecież aby było widać godzinę układ musi wykonywać przynajmniej kilkanaście obrotów na sekundę.

Dlatego postanowiłem zbudować coś odmiennego, co nie wymaga szczotek ani transformatorów obrotowych, a jednocześnie układ elektroniczny jest nieruchomy i w pełni do-

List. 1.

```

Scrstal = 12000000
Config Scl = Portd.1
Config Sda = Portd.0
Config I2cdelay = 10

Config Portb = Output
Portb = 127

Config Int0 = Rising
On Int0 Go
Enable Int0
Enable Interrupts

Dim I As Byte , Ii As Byte
Dim B As Byte
Dim S As Byte , M As Byte , H As Byte
Dim Ustawione As Byte
Dim Znaki(8) As Byte
Znaki(3) = 10           'dwukropek
Znaki(6) = 10         'dwukropek

Declare Sub Get_time
Declare Sub Set_time
Declare Sub Setting_time

Do
  Call Get_time

  If Pind.4 = 0 Then Call Setting_time
  Waitms 100
Loop

Go:
  Waitms 20           'odległość od krawędzi
  Znaki(1) = H / 10
  Znaki(2) = H Mod 10
  Znaki(4) = M / 10
  Znaki(5) = M Mod 10
  Znaki(7) = S / 10
  Znaki(8) = S Mod 10
  For I = 1 To 8
    For Ii = 0 To 4
      B = Znaki(i) * 5
      B = B + Ii
      B = Lookup(B , Dta_znaki)
      Portb = B
      Waitus 400      'SZEROKOŚĆ ZNAKU
    Next Ii
    Portb = 127
  Waitus 400          'ODSTĘP
  Next I
Return

Sub Setting_time
  S = 0
  Do
    Waitms 100
    If Pind.5 = 0 Then
      If Ustawione = 2 Then
        Ustawione = 0
        Call Set_time
        Exit Do
      Else
        Ustawione = 1

```

List. 1. c.d.

```

  Incr H
  If H > 24 Then H = 0
  End If
  End If
  End If
  If Pind.4 = 0 Then
    If Ustawione <> 0 Then
      Ustawione = 2
      Incr M
      If M > 59 Then M = 0
    End If
  End If
  End If
  Loop
End Sub

Sub Get_time
  I2cstart
  I2cwbyte 160
  I2cwbyte 2
  I2cstart
  I2cwbyte 161
  I2crbyte S , Ack
  I2crbyte M , Ack
  I2crbyte H , Nack
  I2cstop
  H = Makedec(h)
  M = Makedec(m)
  S = Makedec(s)
End Sub

Sub Set_time
  M = Makebcd(m)
  H = Makebcd(h)

  Disable Interrupts
  I2cstart
  I2cwbyte 160
  I2cwbyte 0
  I2cwbyte 8
  I2cstop

  I2cstart
  I2cwbyte 160
  I2cwbyte 2
  I2cwbyte S
  I2cwbyte M
  I2cwbyte H

  I2cstop
  Enable Interrupts
End Sub

Dta_znaki:
Data &B1000001 , &B0111110 , &B0111110 , &B0111110 , &B1000001 '0
Data &B1111111 , &B1011110 , &B0000000 , &B1111110 , &B1111111 '1
Data &B1011110 , &B0111100 , &B0111010 , &B0110110 , &B1001110 '2
Data &B0111101 , &B0111110 , &B0101110 , &B0010110 , &B0111001 '3
Data &B1110011 , &B1101011 , &B1011011 , &B0000000 , &B1111011 '4
Data &B0001101 , &B0101110 , &B0101110 , &B0101110 , &B0110001 '5
Data &B1100001 , &B1010110 , &B0110110 , &B0110110 , &B1111001 '6
Data &B0111111 , &B0111000 , &B0110111 , &B0101111 , &B0011111 '7
Data &B1001001 , &B0110110 , &B0110110 , &B0110110 , &B1001001 '8
Data &B1001111 , &B0110110 , &B0110110 , &B0110101 , &B1000011 '9
Data &B1111111 , &B1001001 , &B1001001 , &B1111111 , &B1111111 ':
```

stępny w czasie działania urządzenia. Opisany przeze mnie zegar widmowy wyświetla godzinę za pomocą diod umieszczonych na plastikowym ramieniu drgającym w ruchu wahadłowym.

Program

Program mikrokontrolera przedstawia **List. 1.** Napisany został w znanym i prostym języku Bascom. Zastosowany mikrokontroler to bardzo popularny AT90S2313, którego obecnie produkowanym odpowiednikiem jest Attiny-2313. Ogólnie rzecz biorąc zadaniem mikroprocesora jest odczyt czasu z układu RTC i na podstawie otrzymanych informacji zaświecanie odpowiednich diod LED w odpowiednim czasie, oraz oczywiście umożliwienie użytkownikowi ustawienia właściwej godziny. Synchronizacja wyświetlanego obrazu jest realizowana za pomocą przerwania Int0, na które podawany jest sygnał prostokątny sterujący elektromagnesami. Praktycznie połowę czasu mikroprocesor zużywa na obsługę tego przerwania, natomiast druga połowa (czyli czas gdy wahadło przemieszcza

się ze strony prawej do lewej) przeznaczona jest na odczyt czasu z układu RTC oraz kontrolowanie przycisku powodującego wejście w tryb ustawiania godziny.

Program składa się kolejno z części deklaracji i konfiguracji sprzętowej procesora (której dokładniejszy opis będzie chyba zbędny), pętli głównej oraz czterech procedur:

Go – przerwanie realizujące wyświetlenie godziny,

Setting_time – tryb ustawiania godziny,

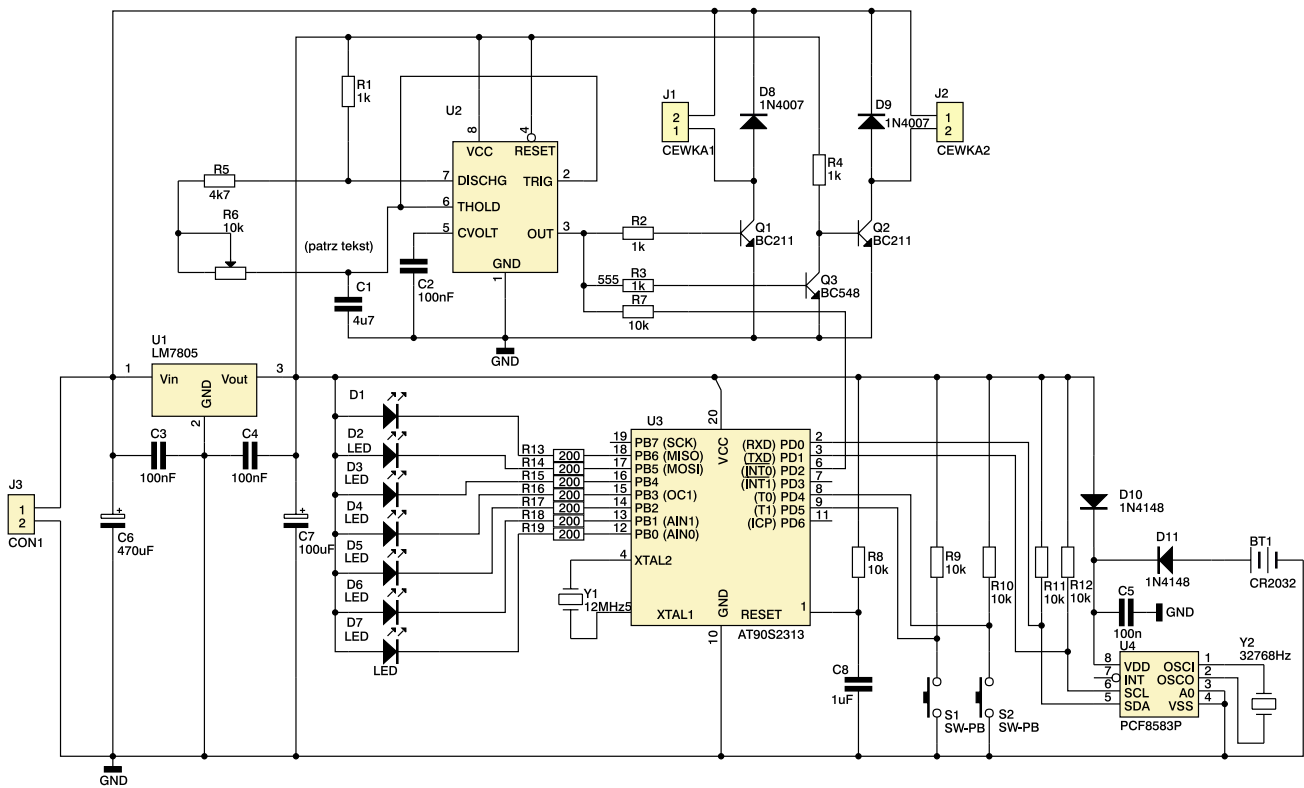
Get_time – realizacja odczytu czasu za pomocą magistrali I²C,

Set_time – realizacja zapisu ustawionego wcześniej czasu, również za pomocą I²C.

Pętla główna dziesięciokrotnie w ciągu sekundy wywołuje procedurę odczytu czasu oraz sprawdza czy przycisk S1 podłączony do pinu d.4 jest wciśnięty. Jeśli tak, to wywołana zostaje procedura ustawiania czasu. Wówczas czas jest zerowany (zostaje wyświetlone 00:00:00) i za pomocą przycisku S2 należy ustawić aktualną godzinę. Następnie, dopiero po ustawieniu go-

dziny, przechodzimy do ustawiania minut za pomocą przycisku S2. Ustawianie sekund uznałem za zbędne, dlatego przy wejściu do trybu ustawiania czasu są one zerowane i aby zegar był punktualny co do sekundy należy ustawić czas o minutę późniejszy niż aktualny, a następnie poczekać aż ta minuta nastanie – w tym momencie należy ponownie wcisnąć przycisk S1, co spowoduje wywołanie procedury zapisu czasu do układu RTC a następnie powrót programu do pętli głównej.

Do tej pory program jest prosty i zrozumiały, jednak procedura **Go** jest trochę bardziej zagmatwana. Komenda *waitms 20* rozpoczynająca procedurę ustala opóźnienie wyświetlania godziny w stosunku do sygnału przerwania, czyli w praktyce odległość wyświetlanego obrazu od lewej skrajnej pozycji wahadła. Godzina powinna być wyświetlana w czasie gdy ruch wahadła jest najbardziej jednostajny, czyli w rejonie jego środkowego położenia, dlatego wartość tego opóźnienia trzeba ustalić indywidualnie do każdej konstrukcji. Kolejne sześć linii



Rys. 1. Schemat elektryczny zegara

procedury zapisuje kolejne cyfry godzin, minut i sekund do tablicy *Znaki*, która zawiera osiem znaków wyświetlanych kolejno przez zegar (dwa dwukropki zostały wpisane do tej tablicy na stałe na początku programu). Następnie program przechodzi do podwójnej pętli *For*. Pętla zewnętrzna wykonuje się osiem razy, ponieważ tyle jest wyświetlanych znaków, zmienna *l* określa który raz pętla jest wykonywana, czyli który znak jest aktualnie wyświetlany. Natomiast pętla wewnętrzna wykonuje się pięć razy dla każdego znaku, ponieważ jeden znak składa się z pięciu pionowych linii, zmienna *li* określa która linia (z *l*-tego znaku) jest w danej chwili wyświetlana.

W tym momencie należałoby wspomnieć, że konfiguracja zaświecenia diod LED zapisana jest w obszarze danych na końcu programu pod etykietą *Dta_znaki*. Zapis ten składa się z 55 liczb ośmiobitowych (dla ułatwienia zapisałem je binarnie), które reprezentują kolejno pionowe linie jedenastu znaków, od „0” do „9” oraz „:”. Kolejne bity liczby reprezentują kolejne diody LED, gdzie „0” oznacza diodę zaświeconą a „1” zgaszoną. Najstarszy, ósmy bit każdej liczby jest nieistotny, ponieważ mamy tylko 7 diod LED. Teraz wróćmy do dalszej analizy programu.

Zmienna *B* przyjmuje wartość reprezentującą miejsce pierwszej pionowej linii znaku zapisanego w tablicy *Znaki* na pozycji *l*-tej, każdy znak to 5 linii dlatego wartość z tablicy mnożona jest przez 5. Następnie wartość *B* zwiększana jest o wartość *li*, czyli o wartość aktualnej linii z aktualnego znaku. Następnie z obszaru danych *Dta_znaki* pobierana jest wartość zapisana na pozycji *B*-tej. Dla przykładu, jeśli jest godzina 14 i w danej chwili trzeba zaświecić trzecią

pionową linię cyfry drugiej (czyli cyfry „4”), to program pobierze informację które diody włączyć z $4 \times 5 + 3 = 23$ pozycji z obszaru danych *Dta_znaki*. Następnie odczytana liczba zostaje wystawiona na port B do którego podłączone są diody LED, po czym program czeka 400 μ s (ta wartość określa szerokość pojedynczego znaku). Na tym kończy się pętla wewnętrzna, i po wyświetleniu wszystkich pięciu linii na port B zostaje wystawiona liczba 127 (binarnie 1111111), czyli wszystkie diody zostaną zgaszone, ponieważ właśnie trwa przerwa między znakami trwająca zgodnie z wartością wpisaną w następnej linii programu 400 μ s. Po odczekaniu tego czasu program przechodzi do wyświetlania następnego znaku, a jeśli wyświetlone zostały już wszystkie z ośmiu to następuje powrót do pętli głównej i oczekiwanie na następny sygnał przzerwania.

Konstrukcja mechaniczna

Ważnym elementem przedstawionego zegara jest konstrukcja mechaniczna widoczna na zdjęciach. Zasadniczym elementem jest tutaj zamocowany pionowo plastikowe ramię stanowiące wahadło, na którego końcu umieszczone są diody LED. Ramię to jest zamocowane sztywno do płyty PCV będącej podstawą urządzenia. W płycie wycięta jest szczelina, w którą koniec ramienia jest wsunięty i przymocowany od spodu za pomocą dwóch aluminiowych kątowników. Podstawa jest dwupoziomowa, aby między płyty można było wsunąć ciężarki. Zegar z uwagi na zasadę działania wymaga balastu, aby po włączeniu nie zaczął „tańczyć” po stole. Ja jako tego balastu użyłem dwóch otwieranych odważników do nurkowania o łącznej masie

3 kg. Tyle w zupełności wystarczy aby zegar stał stabilnie. Warto jest wyposażyć dolną płytę w jakieś gumowe nóżki, dzięki czemu cała konstrukcja podczas pracy będzie stała jeszcze stabilniej. Do górnej płyty podstawy przymocowana jest płytka drukowana z układem elektronicznym, do jej zamocowania użyłem sześciokątnych kołków dystansowych, które służyły wcześniej do utrzymywania płyty głównej w obudowie komputera PC. Kołki te wkręciłem w wywiercone wcześniej, nieco węższe

WYKAZ ELEMENTÓW

Rezystory
 R1...R4: 1 k Ω
 R5: 4,7 k Ω
 R6: 10 k Ω pot. montażowy wielobrotowy
 R7...R12: 10 k Ω
 R13...R19: 200 Ω

Kondensatory
 C1: 4,7 μ F (patrz tekst)
 C2...C5: 100 nF
 C6: 470 μ F
 C7: 100 μ F
 C8: 1 μ F

D1...D7: LED zielona
 D8, D9: 1N4007
 D10, D11: 1N4148
 T1, T2: BC211
 T3: BC548
 U1: LM7805 w obudowie TO-220
 U2: NE555
 U3: AT90S2313 (ATTiny2313)
 U4: PCF8583P

Inne
 Y1: Rezonator kwarcowy 12 MHz
 Y2: Rezonator kwarcowy „zegarkowy” 32,768 kHz
 BT1: Bateria CR2032
 S1, S2: Microswitch
 J1...J3: Złącza



niż średnica gwintu kołków otwory w płycie, dzięki czemu nic od spodu nie wystaje i nie trzeba używać nakrętek. Do ramienia przymocowane są po bokach dwa magnesy neodymowe (ja zastosowałem magnesy z dysków twar-dych), a przy magnesach znajdują się cewki odpowiedzialne za rozbijanie ramienia. Cewki te w oryginalnym projekcie zostały wyjęte ze starych liczników telefonicznych, jednak z powodzeniem można zastosować cewki z większych gabarytowo przekaźników na 12 V. Elektromagnesy zostały przymocowane do płyty podstawy za pomocą kątowników aluminiowych. Dobrze jest zadbać o łatwość regulacji odległości rdzeni elektromagnesów od magnesów zamocowanych na ramieniu, gdyż pozwoli to na regulację amplitudy drgań ramienia. Musi być ona na tyle duża, żeby wyświetlany obraz nie musiał być zbyt wąski, ale z drugiej strony na tyle mała żeby ramię nie wygięło się za mocno i nie przykleiło magnesem do rdzenia cewki lub nie zламаło.

Ramię zegara musi poruszać się z częstotliwością swojego rezonansu mechanicznego, dlatego częstotliwość pracy generatora jest zależna od wymiarów oraz materiału, z którego ramię jest zrobione, co skutkuje koniecznością dobrania częstotliwości indywidualnie do wykonanej konstrukcji. Przy doborze wymiarów ramienia i materiału z którego będzie zrobione należy pamiętać o tym, że jego częstotliwość drgań powinna być większa niż 15 Hz aby nie było efektu migotania wyświetlanego obrazu.

Montaż płytki drukowanej

Prezentowany model zegara został wykonany na płytce uniwersalnej, dlatego brak w artykule wzoru ścieżek. Wymiary zastosowanej płytki to 65×85 mm. W przypadku konstruowania układu na płytce uniwersalnej najpierw należy ustawić na niej luźno kluczowe elementy (układy scalone, elementy o dużych gabarytach) w odpowiednich miejscach, a następnie z grubsza zaplanować połączenia które trzeba będzie wykonać. Należy pamiętać, że złącza powinny znajdować się na krawędziach płytki, oraz o tym że na rogach musi znaleźć się miejsce na otwory pod śruby mocujące. Zastosowany w zegarze procesor posiada interfejs ISP (In System Programming), pozwalającą programować go bez wyjmowania go z podstawki

w urządzeniu, więc jeśli dysponujemy takim programatorem warto przy projektowaniu połączeń na płytce uwzględnić złącze ISP (nie ma go na schemacie). Rozmieszczenie elementów i połączeń można zaplanować do końca ze szczegółami, ale ja osobiście po takim zgrubnym zaplanowaniu zaczynam montaż płytki i w trakcie montażu rozwiązania kolejnych połączeń same się pojawiają. Układy scalone, zwłaszcza mikrokontroler należy zamontować w podstawkach.

Po zmontowaniu układu pierwszą czynnością jaką należy zrobić jest sprawdzenie czy generator NE555 działa poprawnie. To jest bardzo prosta część układu więc nikt nie powinien mieć problemu z jego uruchomieniem. Jeśli już pracuje, należy przy pomocy potencjometru ustawić częstotliwość tak, aby ramię drgało jak najmocniej. Jeśli przy pomocy potencjometru nie znajdziemy częstotliwości rezonansu ramienia to może okazać się konieczna wymiana kondensatora C1 na inną pojemność. Po ustawieniu odpowiedniej częstotliwości należy ustawić elektromagnesy w takiej pozycji, aby amplituda drgań była odpowiednia, o czym napisałem wcześniej.

Po uzyskaniu takiego stanu należy włożyć mikrokontroler w podstawkę i wgrać skompilowany wcześniej program, lub jeśli nie korzystamy z ISP program należy wgrać wcześniej. Po włączeniu urządzenia diody powinny już coś wyświetlać, jednak może to być nieczytelne z powodu niedobranego odstępu czasowych. Wówczas należy samodzielnie dobrać wartości opóźnień związanych z odległością obrazu od skrajnej pozycji wahadła, szerokością wyświetlanego znaku oraz odstępu między znakami a następnie po ponownym skompilowaniu programu wgrać go do pamięci mikroprocesora.

Ta konstrukcja może oczywiście wyświetlać nie tylko czas ale również dowolny tekst, to jedynie kwestia programowa. Dlatego zachęcam do samodzielnego eksperymentowania z modyfikacją programu. Znaki są zapisane w takiej samej postaci jak w wyświetlaczach alfanumerycznych LCD (5×7 pikseli) więc jeśli chodzi o litery i inne znaki alfanumeryczne nie trzeba ich wymyślać po pikselu tylko można znaleźć ich gotowe rysunki w notach katalogowych wyświetlaczy.

Karol Łuszcz

R E K L A M A

Miniaturowy wzmacniacz mocy audio AVT1498

Dostępne wersje:

- A - płytka drukowana: 4zł
- B - komplet elementów: 10zł
- C - układ zmontowany: 16zł

www.sklep.avt.pl

ALL YOU NEED!



LEDs

PicoLED™ ExeLED

- Äußerst wettbewerbsfähige, lückenlose Palette in sämtlichen Farben
- Fertigung erfolgt zu 100% in eigenen, kapazitätsstarken Produktionsstätten mit höchstem Qualitätsniveau
- Die PicoLED besitzt mit einer Höhe von 0,2 mm das flachste 0402-Gehäuse der Welt



ANWENDUNGEN

Ideal geeignet für Consumer-, Industrie- und Automotive-Anwendungen.

www.rohmeurope.com