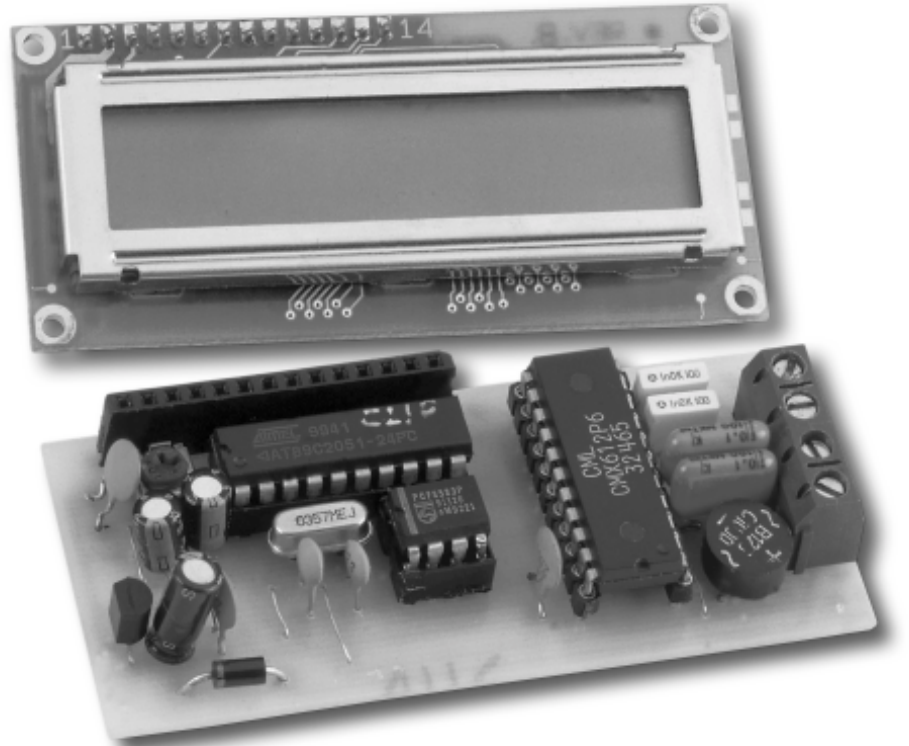


Dekoder CLIP

AVT-5004

Identyfikacja numeru abonenta dzwoniącego CLIP (ang. Calling Line Identification Presentation), ogólnie dostępna w sieciach komórkowych oraz w systemie ISDN, może być stosowana również przez abonentów analogowych. Do tego celu służy prezentowany w artykule dekodery. Program sterujący pracą mikrokontrolera napisano w BASIC-u.



Dekoder działa podobnie jak w telefonach komórkowych: wyświetla numer abonenta dzwoniącego do nas, zanim odbierzemy połączenie. Wiemy więc kto do nas dzwoni i możemy w ten sposób uniknąć niechcianych rozmów. Możemy również zaskoczyć naszego rozmówcę, zwracając się do niego po imieniu zanim się przedstawi. Usługę CLIP możemy uzyskać, jeżeli nasz numer jest numerem z centrali cyfrowej. Aby to sprawdzić, należy skontaktować się z odpowiednim oddziałem BOK (Biuro Obsługi Klienta TP S.A.).

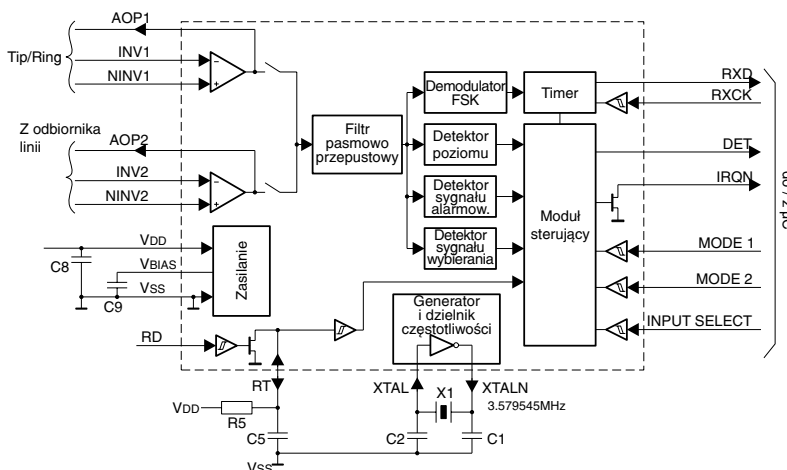
Opis układu

Dane o numerze abonenta wywołującego są wysyłane tuż po pierwszym sygnale dzwonka. Są one przesyłane szeregowo w standardzie V.23 z modulacją FSK. Transmisja danych CLIP charakteryzuje się następującymi parametrami:

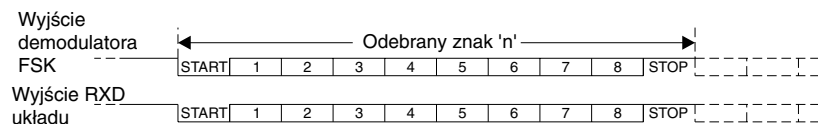
- prędkość transmisji: 1200 b/dów,
- jednokierunkowa transmisja danych, przesyłanych szeregowo asynchronicznie,
- częstotliwości stosowane do kodowania:
 - 1300Hz - poziom logicznej 1
 - 2100Hz - poziom logicznego 0

Dane są wysyłane w postaci meldunku, który składa się z następujących bloków:

- **SMMR** - sygnał ustawiania trybu odbiornika w stan odbioru danych, składa się z sekwencji 300 bitów, których wartość zmienia się naprzemiennie: 0, 1, 0..,
- Sygnał **MARK** - złożony jest z sekwencji 180 bitów, wszystkie mają wartość 1,
- **T1** - bajt określający rodzaj danych, w naszym przypadku będzie miał wartość binarną „10000000” i oznacza identyfikację abonenta dzwoniącego,
- **L1** - bajt określający liczbę bajtów danego meldunku - ten bajt ma zmienną wartość w za-



Rys. 1. Schemat blokowy układu CMX612.



Rys. 2. Współpraca interfejsu RS-232 z demodulatorem FSK.

- leżności od liczby cyfr numeru abonenta dzwoniącego,
- T2 - określa rodzaj danych - czas i data, wartość „00000001“,
- L2 - określa liczbę bajtów potrzebnych do zapisania czasu i daty, L2 ma zawsze wartość „00001000“,
- V2 - osiem bajtów (czas i data) zapisanych w kodzie ASCII,
- T3 - określa rodzaj danych, czyli numer i ma wartość „00000010“,
- L3 - określa liczbę bajtów numeru, ma zmienną wartość w zależności od liczby cyfr numeru abonenta dzwoniącego,
- V3 - numer abonenta zapisany w kodzie ASCII,
- CHECK - jest to bajt sumy kontrolnej (modulo2) wszystkich bajtów meldunku z wyłączeniem samej sumy.

Przykładowa postać meldunku

```

=====
Typ wiadomości 10000000 CLIP
Liczba bajtów 00010101 21 bajtów
Rodzaj danych 00000001 Czas i Data
Liczba bajtów 00001000 8 bajtów
dane 00110000 '0'
dane 00110011 '3'
dane 00110001 '1'
dane 00110101 '5'
dane 00110001 '1'
dane 00110000 '0'
dane 00110011 '3'
dane 00110000 '0'
Data 15 Marzec
Godzina 10:30

Rodzaj danych 00000010 Numer
Liczba bajtów 00001001 9 bajtów
dane 00110000 '0'
dane 00110011 '3'
dane 00110101 '5'
dane 00110001 '1'
dane 00101101 '-'
dane 00110011 '3'
dane 00110010 '2'
dane 00110001 '1'
dane 00110000 '0'
Suma kontrolna 00001110
Numer 0351-3210
    
```

Do odbioru i obróbki sygnału CLIP zastosowano układ firmy CML - CMX612. Jest to scalony odbiornik sygnałów CLIP z modu-

lącją FSK w standardzie V.23. Na rys. 1 przedstawiono jego schemat blokowy. Układ jest przystosowany do zasilania napięciem o wartości od 2,7V i pobiera zaledwie 0,5mA prądu. Za pomocą tego układu można: wykryć sygnał dzwonienia, odebrać dane FSK i przetworzyć je na dane binarne w standardzie RS232. Możliwa jest także prezentacja numeru abonenta oczekującego, tzn. próbującego się dodzwonić podczas prowadzonej przez nas rozmowy telefonicznej. Układ CMX612 może pracować w jednym z czterech trybów, w zależności od stanów na wejściach M1 i M2. Poszczególne tryby pracy zestawiono w tab. 1. W prezentowanym dekodery wykorzystano dwa tryby pracy układu:

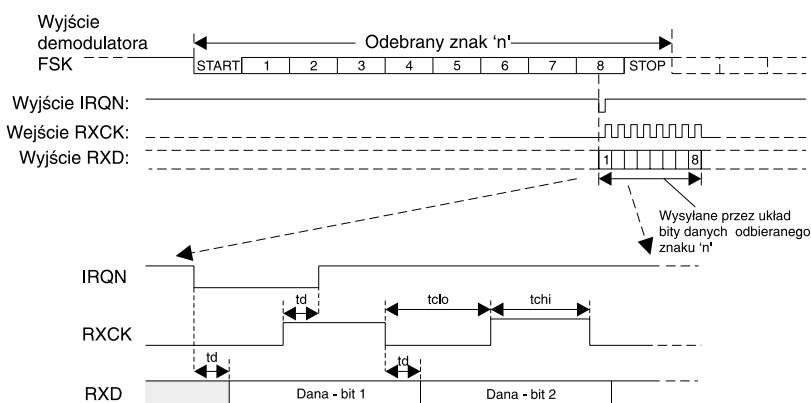
1. Tryb czuwania (M1=1, M2=0) - w tym trybie układ pobiera minimalny prąd, gdyż pracuje tylko detektor dzwonienia, a pozostałe układy wewnętrzne nie są aktywne.
2. Tryb odbioru danych CLIP (M1=0, M2=1) - w tym trybie dane wysyłane przez centralę są odbierane przez układ CMX612, dekodowane i wysyłane szeregowo przez wyjście RXD. Dane te można odebrać w dwojaki sposób:
 - Jeżeli na wejściu RXCL ustawimy stan logiczny jeden, to sygnał CLIP odbierany przez układ jest dekodowany i przesyłany bezpośrednio w postaci binarnej na wyjście RXD. Format danych

M1	M2	Tryb pracy układu CMX612
0	0	Detekcja dzwonienia
0	1	Odbiór danych FSK
1	0	Tryb czuwania "zero power"
1	1	Detekcja tonu oczekiwania

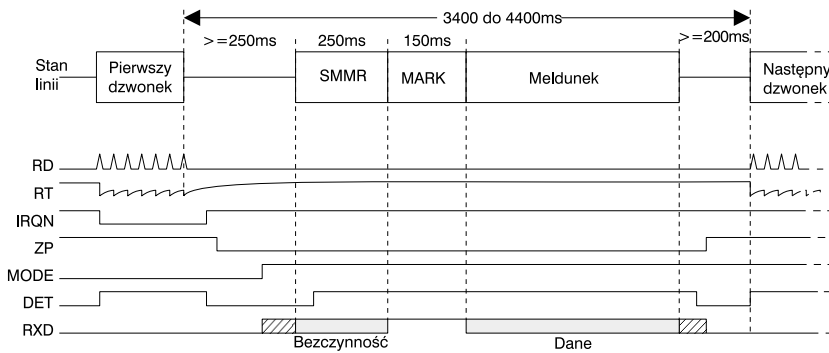
na wyjściu RXD jest zgodny ze standardem RS232, a prędkość przesyłanych danych wynosi 1200b. Przebiegi czasowe przedstawiono na rys. 2. Ten tryb odbioru danych został wykorzystany w dekoderyze.

- Jeżeli wejście RXCL jest w stanie logicznym zero, to po odebraniu 8 bitów układ CMX612 umieszcza je w specjalnym rejestrze „Data Retiming“ i zeruje wyjście IRQN. Stan zera logicznego na wyjściu IRQN informuje procesor, że w rejestrze „Data Retiming“ są nowe dane do odebrania. Następnie procesor na wejście RXCL wysyła osiem impulsów i w takt tych impulsów odbiera dane z wyjścia RXD. Ten tryb transmisji uwalnia nas od stosowania sterownika transmisji szeregowej w procesorze. Dane mogą być odbierane z dowolną prędkością, nie większą jednak niż 1MHz. Przebiegi czasowe sygnałów dla tej transmisji przedstawiono na rys. 3. Natomiast na rys. 4 przedstawiono stany logiczne na wyprowadzeniach układu CMX612 podczas odbioru danych CLIP.

Schemat elektryczny odbiornika CLIP przedstawiono na rys. 5. Zawiera on cztery układy scalone, wyświetlacz LCD i kilkanaście elementów biernych. Układ US1 to procesor typu 89C2051 z wewnętrzną pamięcią programu Flash o pojemności 2kB. Steruje on pra-



Rys. 3. Przebiegi charakterystyczne dla pracy układu CMX612.



Rys. 4. Sposób przesyłania informacji CLIP.

całego dekodera. Współpracujący z mikrokontrolerem układ US2 odbiera sygnał z linii telefonicznej i przekształca go do postaci cyfrowej, a następnie przesyła poprzez interfejs RS232. Elementy: R1, R2, R5, R6, R9, C1, C2, C6, G1 spełniają rolę detektora sygnału dzwonienia, a elementy R3, R4, R8, R10, R11, C3, C4 współpracują z wbudowanym w US2 demodulatorem FSK.

Do poprawnej pracy US2 jest niezbędny sygnał zegarowy o częstotliwości 3,579MHz. Żeby nie stosować dwóch rezonatorów kwarcowych, ten sam oscylator wykorzystano do „napędzania” procesora. Kwarc został umieszczony przy wyprowadzeniach procesora, następnie wyjście oscylatora (nóżka 4 US1) zostało połączone z wejściem zegarowym US2 (nóżka 2).

Zastosowanie rezonatora kwarcowego przy procesorze było konieczne, gdyż US2 w stanie czuwania blokuje pracę wewnętrznego oscylatora, co powodowałoby również zatrzymanie pracy procesora. Prezentowany dekoderek ma również funkcje zegara. Do odliczania czasu zastosowano układ PCF8583.

Zastosowanie zewnętrznego układu zegara uwalnia procesor od precyzyjnego odmierzenia czasu, jego rola ogranicza się tylko do odczytywania danych z układu US3 za pomocą interfejsu I²C. W dalszej części artykułu opisano jak ustawia się ten zegar, bo przecież nie ma żadnych klawiszy i nie można ustawić go ręcznie. Do stabilizacji napięcia zasilającego wszystkie układy zastosowano US4. Jest to miniaturowy stabilizator o napięciu wyjściowym 5V i prądzie 100mA. Odbiornik CLIP

w stanie aktywnym pobiera około 20mA prądu, więc zastosowany stabilizator w zupełności wystarcza do jego zasilania.

Działanie dekodera

Po włączeniu zasilania na wyświetlaczu pojawia się napis „CZEKAM”, po czym program przechodzi do pętli głównej.

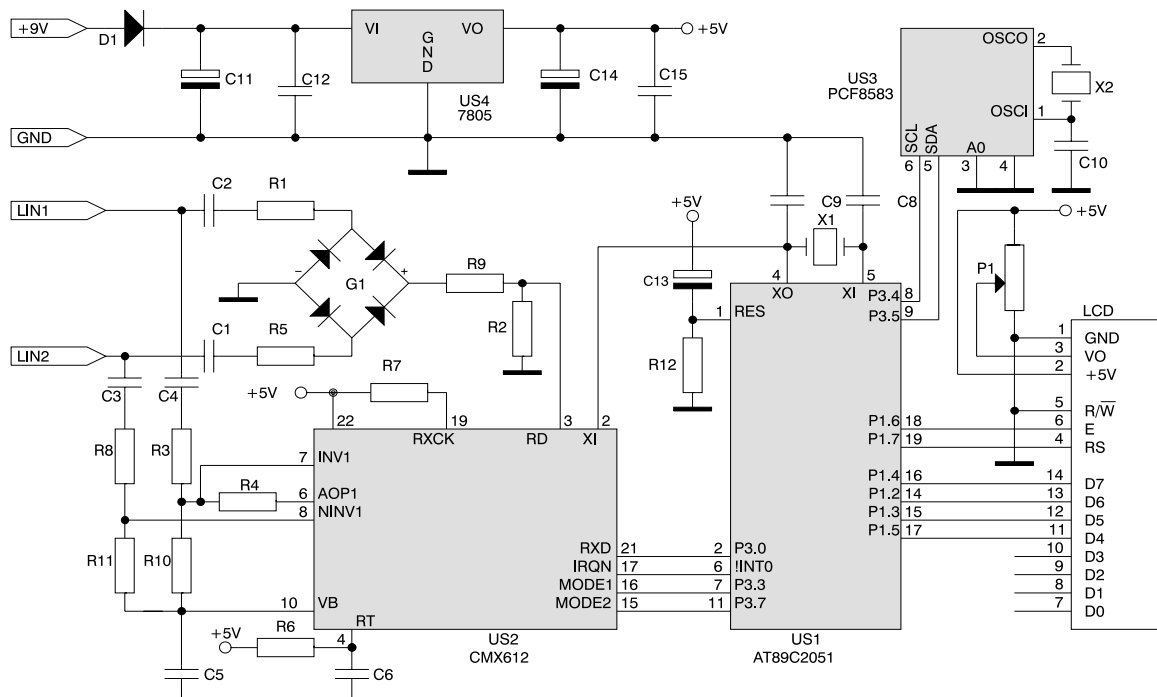
Go:

Do

```

If P3.2 = 0 Then
  Gosub Caller_id
End If
If Stan = 1 Then
  Stan = 0
  Gosub Settime
  Gosub Read_ram
  L = 0
  For T_out = 1 To 30000
    Delay
    If P3.2 = 0 Then
      T_out = 0
    End If
  Next
Do
  If P3.2 = 0 Then
    Gosub Caller_id
    K = 254
  End If
  Waitms 100: Incr K
  If K = 255 Then
    Cls
    Set Cl
    Gosub Gettime
    Goto Go
  End If

```



Rys. 5. Schemat elektryczny dekodera CLIP.

```

Loop
Cls
Set C1
Gosub Gettime
End If
If C1 = 1 Then
Gosub Gettime
End If
Loop
End

```

W pętli głównej jest sprawdzany stan portu P3.2, jeśli P3.2 ma wartość zero, to następuje skok do podprogramu „Caller_id“.

```

Caller_id:
Stan = 0 : T_out = 0
Bitwait P3.2, Set
Waitms 1
Reset P3.3: Set P3.7
Reset P1.1
Waitms 1

Int_5:
F = Inkey
While F <> &B01010101
F = Inkey
Delay
Incr T_out
If T_out = 10000 Then
Enable Int1
Goto Int_end
End If
Wend
T_out = 0

Int_4:
While F <> &B10000000
F = Inkey
Delay
Incr T_out
If T_out = 10000 Then
Enable Int1
Goto Int_end
End If
Wend

Int_3:
Buf(1) = F
F = Waitkey
Buf(2) = F 'liczba bajtów
U = F + 3
For Z = 3 To U
F = Waitkey
Buf(z) = F
Next
Buf(z) = F
Stan = 1
Gosub U2test
If U2 = 0 Then: Stan = 0
End If

Int_end:
Set P3.3: Reset P3.7
Set P1.1
Return

```

Ten podprogram przełącza układ odbiornika CLIP (US2) ze stanu czuwania w tryb aktywny i oczekuje

przez prawie 3 sekundy na pojawienie się danych o wartości „01010101“ na wyjściu RXD układu US2. Jeżeli taka sekwencja nie pojawi się w danym czasie, następuje przełączenie układu US2 w tryb czuwania i powrót do programu głównego. Jeżeli zaś sekwencja „01010101“ się pojawi, to podprogram czeka na odbiór kolejnych bajtów, tzn.: daty, godziny, numeru. Odebrane dane są zapisywane w pamięci RAM w postaci tablicy o nazwie „Buf“. Liczba bajtów tablicy wynosi 25. Ponieważ nie wszystkie bajty meldunku CLIP są przeznaczone do wyświetlenia na wyświetlaczu, zastosowanie tablicy pozwala na łatwe „wybieranie“ tych bajtów, które są w danym momencie potrzebne. Po odebraniu wszystkich bajtów meldunku następuje skok do podprogramu „U2test“.

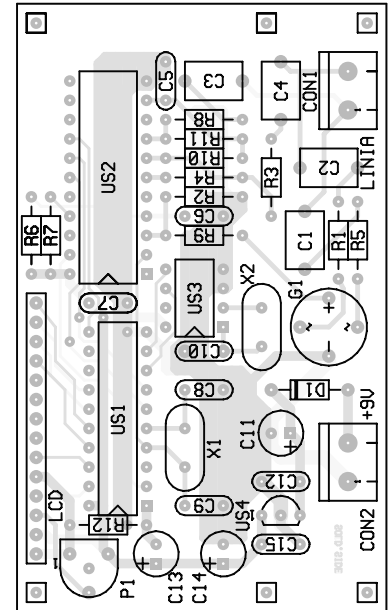
```

U2test:
U = 0
F = Buf(2) + 2
For Z = 1 To F
W = Buf(z)
U = U + W
If U > 127 Then
U = U - 256
End If
Next
U = 256 - U
F = F + 1
If U = Buf(f) Then
U2 = 1
Else
U2 = 0
End If
Return

```

Ten podprogram oblicza sumę modulo 2 (uzupełnienie do 2) wszystkich bajtów meldunku i porównuje z ostatnim bajtem tego meldunku. Podczas wyliczania sumy modulo 2 liczby mniejsze od 127 są traktowane jako dodatnie, a większe jako ujemne. Taki sposób dodawania umożliwia dodawanie nieskończonej liczby bajtów, których suma zawsze będzie jednobajtowa. Jeżeli suma wyliczona przez centralę i przez nasz procesor jest różna, świadczy to o błędach transmisji i numer abonenta dzwoniącego nie może być wyświetlony. Program przechodzi do oczekiwania na następną transmisję.

Jeżeli zaś obydwie sumy są takie same, to następuje ustawienie bitu „Stan“ i powrót do programu głównego. Pojawienie się ustawionego bitu „Stan“ jest dla



Rys. 6. Rozmieszczenie elementów na płycie drukowanej.

programu głównego informacją o poprawnym odebraniu numeru abonenta dzwoniącego i wówczas przechodzi do procedury „Set-time“. Jak wcześniej wspomniano prezentowany odbiornik posiada również zegar i właśnie podprogram „Settime“ jest odpowiedzialny za ustawienie daty i godziny. Ustawienie wszystkich parametrów zegara następuje samoczynnie, wystarczy tylko wysłać sygnał dzwonienia na nasz numer telefonu, np. z telefonu komórkowego i nasz zegar jest już ustawiony. W ten sposób otrzymaliśmy rodzaj zegara DCF, z tą tylko różnicą, że nie jest zsynchronizowany z atomowym wzorcem czasu, lecz z centralą telefoniczną.

Ponieważ dane o dacie i godzinie zapisane są w tablicy „Buf“ w kodzie ASCII, a układ zegara US3 wymaga danych w kodzie BCD, należało dokonać konwersji tych kodów. Ta procedura została przygotowana w podprogramie „Settime“ w assemblerze.

```

H = Buf(9)
M1 = Buf(10)
$asm
mov a, {m1}
anl a, #&b00001111
mov {m1}, a
mov a, {h}
anl a, #&b00001111
Swap A
add a, {m1}
mov {h}, a
$end Asm

```

Zadaniem tej procedury jest pobranie dwóch bajtów z tablicy „Buf” oznaczających dziesiątki godzin i jedności godzin i umieścić je w komórce pamięci „H” (czyli godzina) w postaci dwóch liczb BCD. Podprogram musi być wykonany dla wszystkich bajtów daty i godziny. Następnie dane o dacie i godzinie w kodzie BCD są umieszczone w komórkach o nazwach odpowiednio: H-godziny, M-minuty, D-dni, Month-miesiące. Tak przetworzone dane są wysyłane do układu US3 magistralą I²C w następującej postaci:

```
I2Cstart
I2Cwbyte &HA0
I2Cwbyte 0
I2Cwbyte 8
I2Cstart
I2Cwbyte &HA0
I2Cwbyte 2
I2Cwbyte S
I2Cwbyte M
I2Cwbyte H
I2Cwbyte D
I2Cwbyte Month
I2Cstop
```

Po tych czynnościach układ zegara zaczyna odliczanie czasu z nowymi parametrami, a program główny wykonuje skok do podprogramu wyświetlenia numeru abonenta dzwoniącego „Read_ram”.

```
Read_ram:
Deflcdchar 2,254,240,240,248,
            240,240,254,226
Deflcdchar 0,228,255,226,228,
            232,240,255,224

Cls
W = Buf(14)
'liczba cyfr numeru
Z = W : W = 17 - Z
Locate 1 , W : Ram = 15
For U = 1 To Z
    W = Buf(ram)
    Incr Ram
    If W = 80 Then
        Cls
        Lcd "ZASTRZE";Chr(0);"ONY"
        Return
    ElseIf W = 79 Then
        Cls
        Lcd "NIEDOST";Chr(2);"PNY"
        Return
    Else
        Lcd Chr(w)
    End If
Next
W = Buf(2)
For U = 2 To W
    Next
Return
```

Na początku tego podprogramu następuje zapisanie do pamięci wyświetlacza LCD polskich liter „ż” i „ę” instrukcją „Deflcdchar”. Będą one przydatne w dalszej części programu w wyświetlanych komunikatach. Następnie jest sprawdzany pierwszy bajt numeru abonenta dzwoniącego, który jest zapisany w tablicy „Buf(15)”. Jeżeli ten bajt ma wartość 80, oznacza to, że dany numer jest zastrzeżony i nie można go wyświetlić i na wyświetlaczu pojawia się napis „ZASTRZEŻONY”. Jeżeli zaś pierwszy bajt ma wartość 79, oznacza to, że abonent dzwoniący do nas jest abonentem centrali analogowej i prezentacja jego numeru jest niemożliwa, gdyż centrale analogowe nie posiadają funkcji CLIP. W tym przypadku na wyświetlaczu zostanie wyświetlony napis „NIEDOSTĘPNY”. Jeśli pierwszy bajt numeru jest różny od 79 lub 80, to numer abonenta dzwoniącego zostaje wyświetlony i następuje powrót do programu głównego.

Numer telefonu jest wyświetlany przez około 30 sekund, następnie na wyświetlaczu pojawia się aktualny czas i data, który jest odczytywany z układu US3 za pomocą podprogramu „Gettime”.

```
Gettime:
Dim Dum As Byte
I2Cstart
I2Cwbyte &HA0
I2Cwbyte 2
I2Cstart
I2Cwbyte &HA1
I2Crbyte S , Ack
I2Crbyte M , Ack
I2Crbyte H , Ack
I2Crbyte Yd, Ack
I2Crbyte Wm, Nack
I2Cstop
Home
Lcd Bcd(h); ":"; Bcd(m);
":"; Bcd(s); " "
Lcd Bcd(yd); "-"; Bcd(wm)
Return
```

Od tego momentu procesor odczytuje czas oraz sprawdza stan linii P3.2. Jeśli pojawi się zero, to od początku zaczyna się procedura odbioru numeru abonenta dzwoniącego.

Montaż i uruchomienie

Układ został zmontowany na płytce jednostronnej o wymiarach płytki wyświetlacza LCD. Jej schemat montażowy przedstawiono na

WYKAZ ELEMENTÓW

Rezystory

R1, R3, R5: 470kΩ SMD
R2: 68kΩ SMD
R4: 680kΩ SMD
R6, R8, R9: 470kΩ SMD
R7, R12: 10kΩ SMD
R10: 200kΩ SMD
R11: 160kΩ SMD
P1: 10kΩ

Kondensatory

C1, C2: 100nF/100V
C3, C4: 1nF/100V
C5, C7, C12, C15: 100nF
C6: 330nF
C8, C9: 33pF
C10: 27pF
C11: 47μF/25V
C13, C14: 10μF/25V

Półprzewodniki

D1: 1N4004
G1: mostek prostowniczy 1A/400V
US1: AT89c2051 zaprogramowany
US2: CMX612
US3: PCF8583
US4: 78L05

Różne

X1: kwarc 3,579MHz
X2: kwarc 32,768KHz
CON1, CON2: ARK2(5mm)
LCD: wyświetlacz LCD 1x16a

rys. 6. Montaż zaczynamy od rezystorów. Ze względu na małe wymiary płytki konieczne było zastosowanie rezystorów wykonanych w technologii SMD. Montaż tych elementów wymaga dużej staranności, ale można go wykonać za pomocą lutownicy transformatorowej. Następnie montujemy podstawki pod układy scalone, kondensatory i na końcu złącza CON1 i CON2.

Do złącza CON2 podłączamy zasilacz o napięciu około 9V, a do złącza CON1 linię telefoniczną. Następnie potencjometr P1 ustawiamy tak, aby uzyskać jak najlepszy kontrast wyświetlacza LCD. Układ zmontowany ze sprawnych elementów działa od razu bez żadnych dodatkowych regulacji.

Krzysztof Pławiuk
krzysztof.plawiuk@ep.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/marzec01.htm> oraz na płycie CD-EP03/2001B w katalogu PCB.