

W tym odcinku kończymy temat dołączania klawiatury do systemu mikroprocesorowego. Zajmujemy się dołączeniem klawiatury matrycowej.

W poprzednim odcinku pokazaliśmy jak sterować klawiaturą składającą się z nie więcej niż 4 klawiszy. Zastosowaliśmy najprostszy sposób jej dołączenia, czyli jeden klawisz zajmował jedną linię portu. Tą metodą liczbę klawiszy można powiększać aż do wyczerpania linii portów. Przez zajęcie wszystkich linii oczywiście traci sens użycie systemu mikroprocesorowego jako urządzenia sterującego, bowiem nie wyda ono żadnych sygnałów sterujących.

Jak pokazaliśmy, efektywna obsługa klawiatury jest zjawiskiem bardzo rzadkim. Jeśli nawet naciskamy klawisz dwa razy na sekundę, to między naciśnięciami mikroprocesor wykona kilkaset tysięcy cykli rozkazowych. Niewiele więc się zmienia, kiedy czas wykonywania procedury obsługi klawiatury wydłuży się nawet dziesięciokrotnie. Procedurę tę możemy komplikować aż do granic wyobraźni, jednak zawsze rozsądnie.

Kiedy obsługujemy więcej niż osiem klawiszy, zaczyna brakować linii jednego portu. W tamtym ustawieniu klawisze tworzyły jedną linię włączników. Wzbogacamy więc naszą koncepcję podłączenia klawiszy dodając drugi wymiar i tworzymy matrycę 4x4 klawiszy (rys. 4). Cztery młodsze linie są połączone z kolumnami, a cztery pozostałe - z wierszami. Na przecięciu linii kolumny z linią wiersza postawiono włącznik, czyli klawisz. Jak w poprzednich przykładach, również do sygnalizacji naciśniętego klawisza posłużymy się czterema zbuforowanymi diodami LED, dla odmiany sterowanymi dodatkowym portem zewnętrznym U4 kitu AVT-222 (EP 11/94). Na ten port będzie wysyłany kod binarny klawisza w czasie, gdy on pozostaje naciśnięty.

Poprzednio odczyt klawiatury polegał na odczycie stanu całego portu P1 i wnioskowaniu o naciśniętym klawiszu. Teraz nie możemy tak zrobić. W układzie z rys. 4 odczyt stanu portu P1 będzie w istocie odczytem stanu jego przerzutników wyjściowych. Klawiatura nie ma bowiem przyłączonego źródła napięcia wymuszającego zmianę tego stanu. Występuje tutaj dodatkowo pewna niedogodność charakterystyczna dla pracy z portem dwukierunkowym. Przypo-

Programowanie '51, cd. Dołączanie klawiatury, część 2

mniej zatem budowę portów P1 i P3. Pokazano to na rys. 5. Na wyjściu każdego z pinów tych portów jest układ z otwartym drenem, co sugeruje funkcję logiczną ZWARTE-I (wired AND, I przewodowe). Ażeby był możliwy odczyt stanu pinu danego portu, na wyjściu przerzutnika musi panować stan wysoki, w przeciwnym razie zawsze będzie to stan niski.

Do odczytu klawiatury matrycowej zastosujemy metodę wybierania poszczególnych kolumn i odczytu stanu wierszy. Na rys. 6 przedstawiono przebiegi czasowe pokazujące realizację tej metody. Na jedną linię portu wystawiamy zero, zaś na pozostałe jedynki. W taki sposób wybieramy jedną kolumnę, czyli wykonujemy to na liniach P1.0-P1.3. Jak wcześniej powiedzieliśmy, do przerzutników bitów P1.4-P1.7 musimy wpisać jedynki, po to aby zapewnić poprawny odczyt stanu portu. Bity te będą odczytywane w następnej fazie procedury obsługi klawiatury. Brak stanu niskiego na żadnym z bitów P1.4-P1.7 implikuje wystawienie zera na kolejnej kolumnie i ponowny odczyt stanu wierszy. Listing 1 przedstawia zapis kodu programu wykonującego odczyt klawiatury, jej zdekodowanie oraz transmisję wyniku na diody LED.

Podprogram opóźnienia DEL_51MS jest znany z poprzedniej części artykułu. Skanowanie i dekodowanie klawiatury przeprowadza się w procedu-

rze obsługi timera T0. Dekodowanie polega tu na wnioskowaniu o kodzie klawisza na podstawie wybranej kolumny i wykrytego niskiego stanu wiersza matrycy, zapisanego w akumulatorze. Kod klawisza jest zapisywany do komórki pamięci RAM o nazwie KEYB, #0EH. Jeśli nie został naciśnięty jakikolwiek klawisz, to zmienna KEYB przyjmie wartość 0FFH. Przedstawiona wyżej procedura jest jedną z ogólniejszych. Zamiast proponowanych wartości możemy przyjąć inne, np. kody znaków ASCII i nie sugerować się strukturą klawiatury.

Sekwencja rozkazów

```
JB ACC.7, TO_S12
MOV KEYB, #0EH
SJMP TO_S3
```

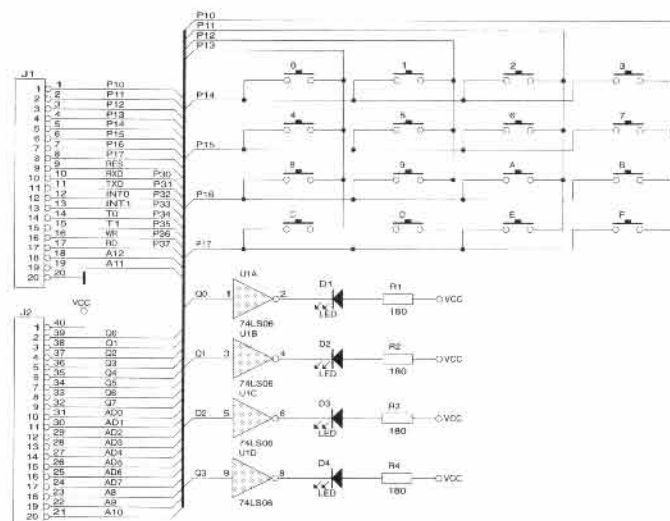
TO_S12:

```
SJMP TO_S22
```

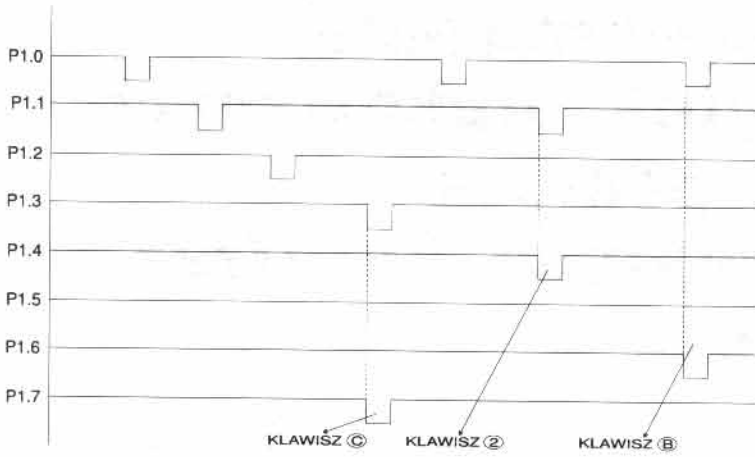
wynika z ograniczonego zasięgu skoków warunkowych. Skoki te są skokami względnymi „w promieniu” 127 bajtów wokół miejsca jego wystąpienia.

Program główny jest kolejnym poziomem interpretacji informacji przychodzącej z klawiatury. Tutaj też likwidowane są skutki drgań styków klawiszy.

Jeżeli zachowamy wartości klawiszy z rys.4 oraz regularność struktury matrycy, procedura może się uprościć. Wystarczy zauważyć, że wybierając po kolei wiersze i odczytując kolejno kolumny, można zliczać liczbę tych operacji i przyjmować ją za kod bieżąco odczytywanego klawisza.



Rys. 4. Dołączenie klawiatury matrycowej 4x4 do modułu AVT-22



Rys. 6. Przebiegi czasowe występujące na pinach portu P1 podczas odczytu klawiatury matrycowej z rys. 4.

```

; OBSLUGA PRZERWANIA TO
TO_SERVIS:
    PUSH ACC
    PUSH R0REG
    PUSH R1REG
    PUSH R2REG
    :
    ustawianie liczników timera TO
    :
TO_S1:
    MOV R1,#4
    MOV KEYB,#10H
    MOV R2,#07FH
TO_S6:
    MOV R0,#4
    MOV P1,R2REG
    MOV A,P1
    ORL A,#0F0H
TO_S4:
    RRC A
    DEC KEYB
    JNC TO_S2
    DJNZ R0,TO_S4
TO_S5:
    MOV A,R2
    RR A
    MOV R2,A
    DJNZ R1,TO_S6
    MOV KEYB,#0FFH
TO_S2:
    POP R2REG
    POP R1REG
    POP R0REG
    POP ACC
    RETI
    
```

Listing 2

po którym będzie generowana seria zmian oraz sprawdzany jest stan klawiatury. Po odliczeniu tego czasu rozpoczyna się cyklicznie co ok. 0.2s wykonywanie podprogramu KEY_U4 (od etykiety MAIN9: do MAIN5:). Trwa to tak długo, aż zostanie wykryty kod OFFH sugerujący zwolnienie klawiatury. Jeśli było to zakłócenie, nastąpi powrót do miejsca pierwszego wykrycia naciśniętego klawisza (etykieta MAIN1:)

Sposoby odczytu i interpretacji klawiatury można mnożyć. Wiele zależy od konkretnej aplikacji. Tutaj pokazaliśmy tylko podstawowe i najczęściej spotykane sytuacje w praktyce mikroprocesorowca. Być może niektórzy z Czytelników poczują w tym momencie niedosyt wiedzy z powodu tak małej liczby „gotowców”. Nie możemy i nie chcemy ich wyřęcać, ale życzymy im sukcesów w tworzeniu własnych opracowań.

Mirosław Lach

```

U4 EQU KEYB+1
;PROGRAM KLAWIATURY PC
MAIN:
    MOV A,KEYB
    CJNE A,#0FFH,MAIN1
    SJMP MAIN
MAIN1:
    ACALL KEY_U4
    MOV R3,#5
MAIN2:
    ACALL DEL_50MS
    MOV A,KEYB
    CJNE A,#0FFH,MAIN3
    SJMP MAIN4
MAIN3:
    DJNZ R3,MAIN2
MAIN9:
    MOV R3,#4
MAIN6:
    ACALL DEL_50MS
    MOV A,KEYB
    CJNE A,#0FFH,MAIN5
    SJMP MAIN4
MAIN5:
    DJNZ R3,MAIN6
    MOV A,KEYB
    CJNE A,#0FFH,MAIN7
    SJMP MAIN4
MAIN4:
    ACALL DEL_50MS
    MOV A,KEYB
    CJNE A,#0FFH,MAIN1
    ACALL DEL_50MS
    MOV A,KEYB
    CJNE A,#0FFH,MAIN4
    SJMP MAIN
KEY_U4:
    JB ACC.0,KU4
    INC U4
    MOV A,U4
    MOVX @R0,A
    RET
KU4:
    DEC U4
    MOV A,U4
    MOVX @R0,A
    RET
DEL_50MS:
    MOV R1,#50
DEL_R1_MS:
    MOV R0,#200
DEL_R1_1:
    NOP
    NOP
    DJNZ R0,DEL_R1_1
    DJNZ R1,DEL_R1_1
    RET
    
```

Listing 3