

Autorzy:

Maciej Ilów

Kinga Foksińska

# Podstawy techniki mikroprocesowej

## Sprawozdanie nr 6

### Obsługa klawiatury matrycowej

*Prowadzący – mgr inż. Antoni Sterna*

*Grupa K02-91m, środa TN, godz. 9:15*

#### 1.Cel ćwiczenia

Celem tego laboratorium było zapoznanie się z podstawami obsługi klawiatury matrycowej przy użyciu mikrokontrolera 8051 oraz napisania odpowiednich kodów umożliwiających odczyt i wyświetlenie stanu klawiszy.

W trakcie wykonywania ćwiczeń skoncentrowaliśmy się na implementacji funkcji umożliwiających aktywowanie i odczyt wybranych wierszy i kolumn klawiatury oraz wyświetlenie stanu wciśniętych klawiszy.

#### 2.Przebieg ćwiczenia

Laboratorium nr 6 składało się z 4 zadań o różnym poziomie trudności.

##### 2.1 Uaktywnienie wybranego wiersza klawiatury

Poniższa funkcja służy do uaktywnienia wybranego wiersza na klawiaturze. Przyjmuje on numer wiersza jako argument wejściowy. Kod rozpoczyna się od przeniesienia numeru wiersza do rejestru R1. Następnie sprawdza, czy numer wiersza jest większy od 3, wykonując operację logiczną AND z wartością 0FCh (1111 1100). Jeśli wynik jest niezerowy, oznacza to, że numer wiersza był większy od 3 i kod przechodzi do etykiety „set\_all”. Jeśli numer wiersza był nie większy niż 3, ustawia początkową wartość R0 na 0111 1111b - ta wartość wskazuje na aktywny zerowy wiersz. Po tym w etykiecie „check\_acc” sprawdza wartość licznika wiersza, który jest przechowywany w akumulatorze. Jeśli licznik jest równy 0, przechodzi do etykiety „set\_P5”, w przeciwnym razie dekrementuje wartość licznika wiersza i na zmniejszonej wartości wykonuje operację rotacji w prawo (rr). Ta operacja powoduje zmianę „aktywnego” wiersza na kolejny. Następnie ponownie przechodzi do etykiety „check\_acc” i powtarza kroki, aż licznik będzie wynosił 0. Jeśli numer wiersza był większy od 3, wyłącza wszystkie wiersze –

ustawia bity 4-7 na wartość 1. Jeśli numer wiersza był nie większy od 3 w etykiecie „set\_P5” przenosi wartość rejestru R0 do rejestru ROWS, co powoduje włączenie pojedynczego wiersza.

```
-----  
; Uaktywnienie wybranego wiersza klawiatury  
;  
; Wejście: A - numer wiersza (0 .. 3)  
-----  
kbd_select_row:  
    mov     R5, A  
  
    anl     A, #0FCh      ; sprawdzenie czy numer wiersza wiekszy od 3 (1111 1100 & (11 / 10 / 01 / 00))  
    jnz     set_all       ; jesli otrzymany wynik jest niezerowy oznacza to, ze w A byla liczba wieksza od 3  
  
    mov     R6, #07Fh     ; przenosimy do R0 wartosc 0111 1111 ktora poczatkowo wskazuje na aktywny zerowy wiersz  
check_acc:  
    mov     A, R5         ; w akumulatorze jest numer wiersza ktory posluzy za licznik  
    jz      set_P5        ; jesli licznik rowny 0 przejdz do ustawienia P5  
  
    dec     A             ; odlozenie zmniejszonego licznika do R1  
    mov     R5, A  
  
    mov     A, R6  
    rr      A             ; obrot o jeden bit w prawo - zmiana "aktywnego" wiersza na kolejny  
    mov     R6, A         ; odlozenie zmodyfikowanej wartosci do R0  
    sjmp    check_acc     ; ponowne sprawdzenie licznika  
  
set_all:  
    orl     ROWS, #0F0h   ; wylaczenie wszystkich wierszy  
    sjmp    end_select_row  
  
set_P5:  
    mov     A, R6  
    anl     ROWS, A       ; wlaczenie pojedynczego wiersza  
  
end_select_row:  
    ret
```

## 2.2 . Odczyt wybranego wiersza klawiatury

Ten kod służy do odczytu wybranego wiersza na klawiaturze. Przyjmuje on numer wiersza jako argument wejściowy i zwraca stan wiersza oraz kod klawisza. Kod wywołuje podprogram "kbd\_select\_row" za pomocą instrukcji lcall, aby aktywować wybrany wiersz klawiatury. Zapisuje stan kolumn (P7) do rejestru R3, aby następnie ustawić "wskaźnik" na kolumnę na wartość 1 w rejestrze R2, co początkowo wskazuje na kolumnę 0 (0001). Ustawia numer aktualnej kolumny na 0 w rejestrze R4 i przechodzi do etykiety "check\_column". W tej etykiecie przenosi wartość "wskaźnika" na aktualnie sprawdzaną kolumnę do rejestru A. Następnie wykonuje operację logiczną AND między wartością w rejestrze A, a wartością stanu kolumn w rejestrze R3. Sprawdza, czy w danej kolumnie występuje stan niski. Jeśli tak, przechodzi do etykiety "save\_result". Jeśli w danej kolumnie nie występuje stan niski, inkrementuje wartość numeru aktualnej kolumny w rejestrze R4 i przesuwą "wskaźnik" na kolejną kolumnę, wykonując operację rotacji w lewo (rl) na wartości w rejestrze A. Następnie przechodzi do etykiety "check\_column", aby ponownie sprawdzić kolejne kolumny, dopóki numer aktualnej kolumny nie osiągnie wartości większej niż 3. Jeśli numer aktualnej kolumny jest większy niż 3, przechodzi do etykiety "end\_read\_row". Tam wyłącza flagę przeniesienia i przechodzi do zakończenia odczytu wiersza, zwracając informację o niewciśniętym klawiszu. Jeśli w etykiecie "check\_column" wykryto stan niski w kolumnie, przechodzi do etykiety "save\_result". Tam ustawia flagę przeniesienia, wskazując na wciśnięty klawisz, a następnie przenosi numer aktualnej kolumny do rejestru A jako kod klawisza.

```

;-----
; Odczyt wybranego wiersza klawiatury
;
; Wejscie: A - numer wiersza (0 .. 3)
; Wyjscie: CY - stan wiersza (0 - brak klawisza, 1 - wcisniety klawisz)
;         A - kod klawisza (0 .. 3)
;-----
kbd_read_row:
    lcall    kbd_select_row ; aktywowanie wiersza

    mov     R1, COLS       ; zapisanie stanu kolumn (P7)
    mov     R0, #1         ; "wskaznik" na kolumnie, który początkowo wskazuje na kolumnie 0 (0001)
    mov     R2, #0         ; nr aktualnej kolumny

check_column:
    mov     A, R0          ; w akumulatorze jest informacja o aktualnie sprawdzanej kolumnie
    anl     A, R1          ; sprawdzenie czy w danej kolumnie jest stan niski
    jz      save_result    ; jeśli stan niski to przechodzi do zwrócenia wyniku

    inc     R2             ; jeśli nie ma stanu niskiego sprawdź kolejną kolumnę
    mov     A, R0
    rl      A              ; przesuwam "wskaznik" na kolejną kolumnę
    mov     R0, A

    mov     A, R2
    anl     A, #0FCh       ; sprawdzenie czy numer kolumny jest większy od 3 (1111 1100 & (11 / 10 / 01 / 00))
    jz      check_column   ; jeśli nie jest większy od 3 sprawdź kolejną kolumnę

    clr     C              ; jeśli nr kolumny większy niż 3 to zwróć informację o niewcisniętym klawiszu
    sjmp    end_read_row

save_result:
    setb    C              ; ustawienie informacji o wciśnięciu klawisza
    mov     A, R2          ; przeniesienie do akumulatora informacji o nr kolumny
end_read_row:
    ret

```

## 2.3 Odczyt całej klawiatury

Ten kod służy do odczytu całej klawiatury. Nie przyjmuje on żadnych argumentów wejściowych i zwraca stan klawiatury oraz kod wciśniętego klawisza. Procedura inicjalizuje rejestr R5 na wartość 0, który będzie przechowywał numer aktualnie odczytywanego wiersza. Następnie przenosi wartość numeru wiersza (R5) do rejestru A i wywołuje podprogram "kbd\_read\_row" za pomocą instrukcji lcall, aby odczytać wybrany wiersz klawiatury. Sprawdza, czy flaga przeniesienia jest ustawiona na 1, co oznacza, że w tym wierszu jest wciśnięty klawisz. Jeśli tak, przechodzi do etykiety "save\_result\_2". Jeśli flaga przeniesienia nie jest ustawiona, inkrementuje wartość numeru wiersza w rejestrze R5 i wykonuje operację AND między wartością w rejestrze A, a wartością 0FCh, aby sprawdzić, czy numer wiersza jest większy od 3 (czyli czy ma wartość 11, 10, 01 lub 00). Jeśli numer wiersza nie jest większy od 3, przechodzi z powrotem do etykiety "read\_row", aby odczytać kolejny wiersz. Jeśli numer wiersza jest większy niż 3, przechodzi do etykiety "end\_kbd\_read". Tam wyłącza flagę przeniesienia i przechodzi do zakończenia odczytu klawiatury, zwracając informację o braku wciśniętego klawisza. Jeśli w etykiecie "read\_row" wykryto wciśnięty klawisz, przechodzi do etykiety "save\_result\_2". Tam przenosi numer klawisza do rejestru R6, przenosi wartość 4 do rejestru B, przenosi numer wiersza do rejestru A i wykonuje operację mnożenia (mul AB), aby obliczyć wartość (numer wiersza \* 4). Następnie dodaje numer klawisza do wartości w rejestrze A i ustawia flagę przeniesienia, wskazując na wciśnięty klawisz.

```

;-----
; Odczyt całej klawiatury
;
; Wejście: CY - stan klawiatury (0 - brak klawisza, 1 - wcisnięty klawisz)
;         A - kod klawisza (0 .. 15)
;-----
kbd_read:
    mov     R3, #0
read_row:
    mov     A, R3          ; nr wiersza
    lcall   kbd_read_row
    jc      save_result_2  ; jeśli flaga ustawiona na 1 to znaczy że w tym wierszu jest wcisnięty klawisz

    inc     R3             ; jeśli nie było wcisniętego klawisza, przejdź do kolejnego wiersza
    mov     A, R3
    anl     A, #0FCh       ; sprawdzenie czy numer wiersza jest większy od 3 (1111 1100 & (11 / 10 / 01 / 00))
    jz      read_row       ; jeśli nr wiersza nie większy od 3 przejdź do czytania wiersza

    clr     C              ; jeśli nr wiersza większy od 3 podaj informacje o braku wcisniętego klawisza
    sjmp    end_kbd_read

save_result_2:
    mov     R4, A          ; przenies do R4 informacje o nr klawisza
    mov     B, #4
    mov     A, R3          ; przenies do akumulatora informacje o nr aktywnego wiersza
    mul     AB             ; A = nr wiersza * 4
    add     A, R4          ; A = (nr wiersza * 4) + nr klawisza
    setb    C              ; ustaw informacje o wcisniętym przycisku
end_kbd_read:
    ret

```

## 2.4 Wyświetlenie stanu klawiatury

Funkcja służy do wyświetlenia stanu klawiatury. Przyjmuje ona stan klawiatury i kod klawisza jako argumenty wejściowe i wykonuje następujące czynności. Początkowo sprawdza, czy flaga przeniesienia jest ustawiona na 0 (brak wcisniętego klawisza). Jeśli tak, przechodzi do etykiety "shut\_leds". Jeśli flaga przeniesienia jest ustawiona na 1, wykonuje operację logiczną XOR między wartością w rejestrze A, a wartością 0FFh, aby ustawić przeciwną wartość bitów. Następnie zeruje najstarszy bit (bit 7) w rejestrze ACC i przenosi wartość z rejestru A do rejestru LEDS, co powoduje włączenie odpowiednich diod LED. Przechodzi do etykiety "end\_kdb\_display", kończąc wyświetlanie stanu klawiatury. Etykieta "shut\_leds" ustawia wartość 0FFh w rejestrze LEDS, co powoduje wyłączenie wszystkich diod LED.

```

;-----
; Wyświetlenie stanu klawiatury
;
; Wejście: CY - stanu klawiatury (0 - brak klawisza, 1 - wcisnięty klawisz)
;         A - kod klawisza (0 .. 15)
;-----
kbd_display:
    lcall   kbd_read
    jnc     shut_leds      ; jeśli brak wcisniętego klawisza przejdź do wyłączenia ledów
    xrl     A, #0FFh       ; ustawienie przeciwnych wartości za pomocą xora
    clr     ACC.7          ; ustawienie najstarszego bitu na 0 (ON)
    mov     LEDS, A        ; włączenie diod
    sjmp    end_kdb_display

shut_leds:
    mov     LEDS, #0FFh    ; wyłączenie diod
end_kdb_display:
    mov     A, #4
    lcall   kbd_select_row ; wyłączenie rzędów
    ret

```

### **3.Wnioski**

Podczas ćwiczeń udało nam się wykonać wszystkie zadania z listy.

Prawidłowe działanie kodu klawiatury wymagało zrozumienia działania rejestru klawiatury, w którym stan klawiszy jest przechowywany w postaci binarnej. Dzięki temu mogliśmy skonfigurować odpowiednie piny mikrokontrolera i manipulować danymi w rejestrze, aby odczytać i wyświetlić informacje o wciśniętych klawiszach.

Podczas tworzenia kodów najbardziej wymagająca okazała się dla nas obsługa różnych przypadków, takich jak brak wciśniętego klawisza, czy błąd, gdy numer wiersza lub kolumny przekracza zakres.