

# **Podstawy techniki mikroprocesorowej lab.**

Prowadzący: Mgr inż. Antoni Sterna (E00-74ap, środa 13:15)

sprawozdanie 3 - 2018.04.04

**Jakub Dorda 235013**  
**Marcin Kotas 235098**

16 maja 2018

L<sup>A</sup>T<sub>E</sub>X

# 1 Wprowadzenie/cel ćwiczeń

Celem ćwiczeń było napisanie funkcji obsługujących wyświetlacz LCD.

## 2 Realizacja funkcji

### 2.1 Zadeklarowane stałe

```
WR_CMD EQU 0FF2CH
WR_DATA EQU 0FF2DH
RD_STATUS EQU 0FF2EH
RD_DATA EQU 0FF2FH
LCD_INIT EQU 038H
LCD_CLEAR EQU 001H
LCD_MODE EQU 00EH
```

Zmienne deklarują ustawienia odpowiednich adresów w pamięci, które mapowane są do obsługi wyświetlacza. Zmienna `MODE` deklaruje tryb działania wyświetlacza (dwie linijki oraz przesuwanie wskaźnika na tekst wraz z pisanie)

### 2.2 Inicjalizacja wyświetlacza

```
init_lcd:
    MOV A, #LCD_INIT
    call lcd_cmd
    MOV A, #LCD_CLEAR
    call lcd_cmd
    MOV A, #LCD_MODE
    call lcd_cmd
    ret
```

Funkcja `init_lcd` uruchamia i czyści wyświetlacz, a następnie ustawia tryb działania.

### 2.3 Wysyłanie rozkazów do wyświetlacza

```
lcd_cmd: ; cmd in ACC
    push ACC
    call wait_busy
    MOV DPTR, #WR_CMD
    pop ACC
    MOVX @DPTR, A
    ret
```

Funkcja `init_cmd` wysyła kod rozkazu z akumulatora na odpowiedni adres. Przed każdym wysłaniem rozkazu następuje sprawdzenie, czy wyświetlacz nie wykonuje w tym momencie innego rozkazu.

### 2.4 Sprawdzanie stanu wyświetlacza

Funkcja `wait_busy` czyta status wyświetlacza z adresu `RD_STATUS`, a następnie sprawdza, czy ustawiona jest bit siódmy (busy flag). Sprawdzanie następuje w pętli aż flaga nie będzie ustawiona.

## 2.5 Wypisywanie znaku na wyświetlacz

```
lcd_writec: ;char in ACC
    push ACC
    call wait_busy
    MOV DPTR, #WR_DATA
    pop ACC
    MOVX @DPTR, A
    ret
```

Funkcja `lcd_writec` pobiera kod znaku z akumulatora, upewnia się, że wyświetlacz nie jest zajęty, a następnie wysyła kod znaku pod adres `WR_DATA`.

## 2.6 Wypisywanie tekstu na wyświetlacz

```
lcd_writestr: ;str pointer in
    DPTR
    clr A
    movc A, @A+DPTR
    jz lcd_writestr_ret
    push dph
    push dpl
    call lcd_writec
    pop dpl
    pop dph
    inc dptr
    jmp lcd_writestr

    lcd_writestr_ret:
    ret
```

Funkcja `lcd_writestr` pobiera adres do tekstu z rejestru `DPTR`, a następnie kopiuje jeden znak do akumulatora. Jeżeli pobrany do akumulatora znak jest równy 0, to cały tekst został już wypisany i funkcja kończy działanie.

W pozostałych przypadkach funkcja odkłada na stos obecny wskaźnik w `DPTR` i wywołuje funkcję `lcd_writec`. Następnie przywraca poprzedni wskaźnik na tekst do `DPTR`, zwiększa go o 1 i wraca na początek funkcji.

## 2.7 Manipulowanie pozycją wskaźnika tekstu

```
    anl A, #00011111B    // x in
    4 lowes bits, y in bit 5
    jnb ACC.4, first_line
    clr ACC.4
    add A, #40H
first_line:
    orl A, #10000000B    //
    set addr command
    lcall lcd_cmd
    ret

wait_busy:
```

Funkcja `lcd_gotoxy` pobiera wybrany adres wskaźnika z akumulatora w następującym formacie: najmłodsze 4 bity oznaczają pozycję x, bit na pozycji czwartej oznacza rząd (0 lub 1). Dzięki temu bity w pierwszym rzędzie adresowane są od 0 do 15, a bity w drugim rzędzie od 16 do 31.

Funkcja najpierw zeruje wszystkie bity 5-7 tak, aby ograniczyć wartość do 31 w przypadku błędnego argumentu. Następnie sprawdza, czy bit czwarty jest ustawiony i jeśli tak, to najpierw go zeruje, a później dodaje do całej wartości 40H. Dzieje się tak ponieważ w rejestrze danych wyświetlacza rząd drugi jest umiejscowiony w adresach od 40H.

Po uzyskaniu końcowego adresu funkcja ustawia bit siódmy, co jest informacją dla wyświetlacza, że młodsze siedem bitów jest adresem wskaźnika pamięci danych. W tej postaci rozkaz przekazywany jest do funkcji `lcd_cmd`.

## 2.8 Zamiana liczb na tekst

```
wait:
movx A, @DPTR
jb ACC.7, wait
ret

text: DB 'qwerty', 0

END
```

Funkcja `int_to_str` pobiera liczbę z akumulatora, a następnie dzieli ją przez 10. Po tej operacji w akumulatorze jest wynik dzielenia, a w rejestrze B jest reszta z dzielenia. Do każdej z tych wartości dodawany jest kod ASCII znaku 0, a następnie funkcja zwraca przez stos oba znaki. Funkcja najpierw odkłada na stos resztę z dzielenia, a następnie wynik, aby po wywołaniu funkcji znaki mogły być pobierane ze stosu i przekazywane do funkcji `lcd_writec` w odpowiedniej kolejności.

## 3 Wnioski/podsumowanie

Wszystkie funkcje zrealizowane podczas zajęć uruchomiły się poprawnie. Funkcja zamiany liczb na tekst została dopisana po zajęciach.