

Wydział Informatyki, Laboratorium Architektury Komputerowej	Data 23.11.2023r.
Ćwiczenie nr.3 Temat: Wprowadzenie do wyświetlacza LCD  Grupa LAB.02 Imię i Nazwisko: Dominik Gąsowski	Prowadzący: Dr inż. Mirosław Omieljanowicz  Ocena:

## Zadanie 1.

### Treść zadania:

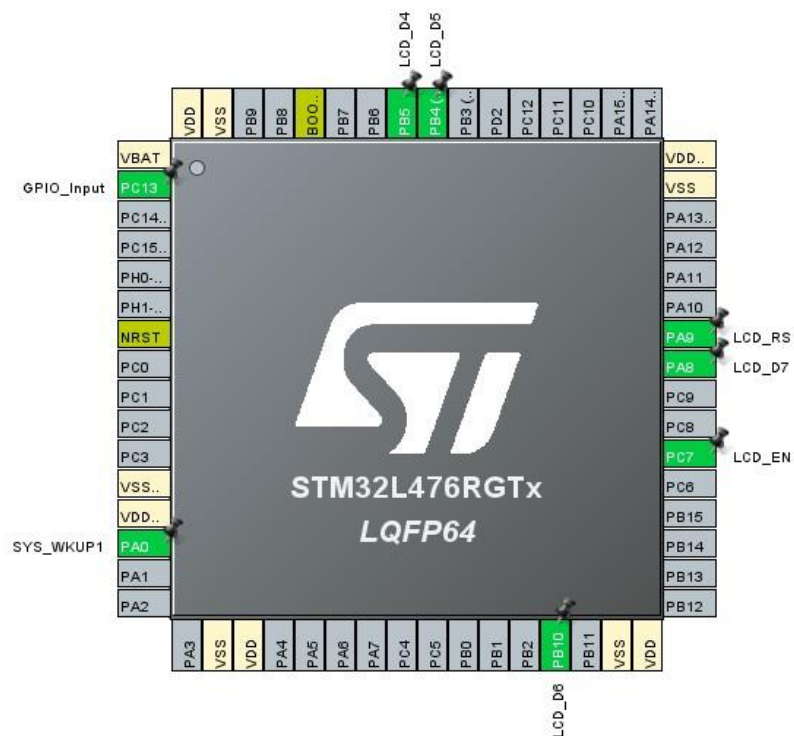
Skonfiguruj piny mikrokontrolera podłączone do sterownika HD44780U w taki sposób, aby możliwe było wyświetlanie znaków na wyświetlaczu LCD. Następnie, w pierwszej linii wyświetl „Grupa”, gdzie oznacza nr twojej grupy laboratoryjnej. W drugiej linii wyświetl imiona członków twojej grupy oddzielone przecinkami.

### Wykonanie:

Na zajęciach nie musieliśmy ręcznie podłączać poszczególnych pinów do gniazd wyświetlacza. Jednak jeśli musielibyśmy, to musimy zrobić to w poprawny sposób zgodny z tabelą:

Nazwa pinu na L476RG	Tryb GPIO	Etykieta na wyświetlaczu
PC7	Wyjście	LCD_EN
PA9	Wyjście	LCD_RS
PB5	Wyjście	LCD_D4
PB4	Wyjście	LCD_D5
PB10	Wyjście	LCD_D6
PA8	Wyjście	LCD_D7

Konfiguracja pinów mikrokontrolera:



```
MX_GPIO_Init();
/* USER CODE BEGIN 2 */
char* message1 = "Grupa 2, Wojtek";
char* message2 = "Dominik, Grzesiek";

// inicjacja wyświetlacza
lcd_init(4, 5*8, 2);
lcd_clear();
// wyświetlenie napisu na pierwszym i drugim wierszu wyświetlacza
lcd_print(1,1, message1);
lcd_print(2,1, message2);
```

Rezultat:



## Zadanie 2.

### Treść zadania:

Napisz program, który będzie cyklicznie przesuwiał dowolny ciąg znaków od lewej krawędzi ekranu do prawej, a następnie od prawej do lewej, tak aby osiągnąć efekt „odbijania” się tekstu. Do zrealizowania zadania wykorzystaj funkcjonalność „display shift” sterownika HD44780U. Ciąg znaków oraz odstępy czasowe pomiędzy przesunięciami ekranu dobierz samodzielnie w taki sposób, aby możliwe było zaprezentowanie poprawności wykonania zadania.

### Wykonanie:

```
// inicjacja zmiennych i tworzenie napisu na LCD
int start = 1, prawo = 1;
int szerokosc = 16;
// napis który wyświetlam
char* message = "Grupa 2";
int poz = 1;
int prawo = 1;
//inicjacja wyświetlacza
lcd_init(4,5*8,2);
lcd_clear();
// wyświetlenie napisu "Grupa 2" w 1 wierszu wyświetlacza
lcd_print(1, 1, message);
```

```

// przesuwanie napisu wokół ekranu
while (1)
{
    if (prawo)
    {
        if (strlen(message) + poz <= 16)
        {
            lcd_cmd(_SHIFT_RIGHT);
            poz++;
            HAL_Delay(600);
        }
        else
            prawo = false;
    }
    else
    {
        if (poz > 1)
        {
            lcd_cmd(_SHIFT_LEFT);
            poz--;
            HAL_Delay(600);
        }
        else
            prawo = true;
    }
}

```

Rezultat:



### Zadanie 3.

#### Treść zadania:

Napisz program, w którym zdefiniujesz dwa własne znaki, następnie używając ich wyświetlisz napis na ekranie, który będzie przesuwając ten znaków do końca pierwszego wiersza, następnie ten sam ciąg pojawi się na końcu drugiego wiersza. Napis następnie będzie przesuwany do początku drugiego wiersza, na koniec będzie się pojawiać znowu na początku pierwszego wiersza.

#### Wykonanie:

```
// funkcja definiująca własny znak specjalny
void lcd_define_custom_char(uint8_t char_num, const unsigned char *data) {
    lcd_cmd(0x40 + (char_num * 8)); // Ustawienie adresu CGRAM dla danego znaku
    for (int i = 0; i < 8; i++) {
        lcd_char_cp(data[i]); // Zapis wzorca do pamięci CGRAM
    }
}
```

```

lcd_init(4, 5*8, 2);

//tablica przedstawiajaca znak specjalny - serce
const uint8_t heart[] = {
    0b000000,
    0b01010,
    0b11111,
    0b11111,
    0b01110,
    0b00100,
    0b00000,
    0b00000
};

```

```

// inicjacja zmiennych
char message[] = "Dominik";
int textLength = strlen(message);
lcd_define_custom_char(1, heart); // definiowanie niestandardowego znaku

// Zamiana litery 'o' na znak serca w zmiennej message
for (int i = 0; i < textLength; i++) {
    if (message[i] == 'o') {
        message[i] = 1; // Ustawienie indeksu niestandardowego znaku serca
        break;
    }
}
int delay=600;

int a = 1;
int ile_wyswietlic = textLength-1;
lcd_init(_LCD_4BIT, _LCD_FONT_5x8, _LCD_2LINE);
lcd_clear();
lcd_define_custom_char(0, heart);

// wyswietlenie napisu
while(a<=textLength)
{
    lcd_clear();
    lcd_print(1, 1, &message[textLength-a]);
    a++;
    HAL_Delay(delay);
}

```

```

while (1)
{
    a = textLength + 1;
    ile_wyswietlic = textLength-1;
    // segment 1 - przesuniecie wyrazu do lewej strony dolnego wiersza
    while(a <= 16)
    {
        lcd_cmd(_SHIFT_RIGHT);
        HAL_Delay(delay);
        a++;
    }

    a = 16 - textLength + 2;

    // segment 2 - przesuniecie wyrazu poza ekran w dolnym wierszu
    while(a <= 16+1)
    {
        lcd_clear();
        lcd_print(1, a, &message);
        lcd_print(2, 16-(textLength-ile_wyswietlic)+1, &message);
        a++;
        ile_wyswietlic--;
        HAL_Delay(delay);
    }

    // segment 3 - przesuniecie wyrazu na gorny wiersz
    // wylonienie go zza ekranu i przesuniecie do prawej strony
    lcd_clear();
    lcd_print(2, 16- textLength + 1, &message);
    a = 16 - textLength;

    while(a > 0)
    {
        lcd_cmd(_SHIFT_LEFT);
        HAL_Delay(delay);
        a--;
    }

    a = 1;

    // segment 4 -- chowanie napisu z prawej strony i wyswietlanie go
    // z powrotem w wierszu nizej z poza ekranu
    while(a <= textLength)
    {
        lcd_clear();
        lcd_print(1, 1, &message[textLength-a]);
        lcd_print(2, 1, &message[a]);
        a++;
        HAL_Delay(delay);
    }
}

```



Rezultat:





## Wnioski:

Wyświetlacz LCD, którym posługujemy się na zajęciach ma 2 wiersze, każdy po 16 bitów. Przesuwając napis w lewą stronę, możemy przesunąć go w nieskończoność pomimo że ostatni indeks wyświetlacza to 16. Po prostu napis będzie nam wychodził poza ekran. Gorzej jest jeśli chcemy, aby nasz napis wyszedł z prawej strony poza ekran. Nie da się go tam wyrzucić, ponieważ minimalny indeks to 0. Nie ma indeksów ujemnych. Rozwiązałem problem wyświetlając napis o jedną literę mniej w każdym obrocie pętli.