

## Wyświetlacz LCD z układem HD44780

Wyświetlacze alfanumeryczne używane podczas ćwiczenia mają dwa wiersze i 16 kolumn. Pojedynczy znak jest wyświetlany na matrycy 5x8 pikseli (5 pikseli szerokości, 8 pikseli wysokości). LCD z matrycami 5x10 pikseli są znacznie rzadziej produkowane.

Notę katalogową do HD44780 można pobrać np. z <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>

HD44780 posiada pamięć DDRAM (Display Data RAM) oraz CGRAM (Character Generator RAM).

Organizacja pamięci DDRAM dla wyświetlaczy dwuwierszowych pokazana jest w nocie katalogowej na rys. 4 str. 11.

Jeżeli kod ASCII litery A zostanie zapisany w DDRAM pod adresem 2, to zostanie ona wyświetlona w trzeciej kolumnie pierwszego wiersza, natomiast zapisanie kodu ASCII litery B w DDRAM do 0x42 spowoduje pokazanie B w trzeciej kolumnie drugiego wiersza. Fabrycznie zdefiniowane znaki wraz z ich kodami podane są w tab. 4 na str. 17. Część sprzedawanych wyświetlaczy zamiast liter japońskich ma znaki pokazane na str. 18.

Pamięć CGRAM umożliwia utworzenie 8 własnych znaków. Mają one kody 0-7 ( w celu wyświetlenia  $n$ -tego zdefiniowanego znaku należy wpisać do DDRAM wartość  $n-1$ . Pokazanie drugiego nastąpi po zapisaniu do DDRAM kodu 1).

Alfanumeryczne wyświetlacze LCD z HD44780 mają następujące wyprowadzenia:

- 1 GND (VSS)
- 2 VCC (VDD)
- 3 V0
- 4 RS
- 5 R/~W
- 6 E
- 7 DB0
- 8 DB1
- 9 DB2
- 10 DB3
- 11 DB4
- 12 DB5
- 13 DB6
- 14 DB7
- 15 A (anoda diody LED podświetlającej LCD)
- 16 K (katoda diody LED)

Linia RW określa, czy wykonywany będzie odczyt (Read) czy zapis (Write) do wyświetlacza. Jeżeli na tej linii jest stan wysoki, to nastąpi odczyt. Stan niski oznacza zapis. Często RW podłącza się na stałe do masy, jeżeli nie są odczytywane żadne informacje z wyświetlacza..

V0 najczęściej podłącza się do środkowego wyprowadzenia potencjometru wpiętego pomiędzy masę i VCC. (potencjometrem reguluje się, aż będą widoczne piksele). Potencjometr należy wetknąć na płytke stykową.

RS – ang. register selector.

Wyświetlacz może wykonywać instrukcje podane w tab.6 na str. 24-25.

Po ustawieniu odpowiednich stanów linii: RS,RW,DB0,DB1,...,DB7 należy zmienić stan linii E z wysokiej na niski. Opadające zbocze na wyprowadzeniu E powoduje rozpoczęcie wykonywania instrukcji zgodnie z tab. 6. Do czasu zakończenia się wykonywania instrukcji nie można wysłać kolejnego polecenia. Maksymalne czasy wykonywania instrukcji są podane w ostatniej kolumnie tab. 6. Aby mieć pewność, że instrukcja się wykonała, należy odczekać odpowiedni czas (najprostszy sposób sterowania) lub sprawdzać stan BUSY FLAG (szybsza komunikacja z wyświetlaczem, ale bardziej skomplikowana). Sposób sprawdzania stanu BUSY FLAG jest przedstawiony na rys. 15 ze str. 32.

Do sterowania wyświetlacza można wykorzystać osiem wyprowadzeń DB0..DB8 i linie sterujące RS,RW,E (interfejs 8-bitowy). Wówczas wykorzystywanych jest aż 11 wyprowadzeń. Druga możliwość polega na użyciu tylko czterech wyprowadzeń DB7-DB4 wraz z RS,RW,E (interfejs 4bitowy - wystarczy 7 linii jednego portu STM32). Komunikacja za pomocą interfejsu 4bitowego została przedstawiona na rys.9 na str. 22. (najpierw są wysyłane 4 starszy bity a następnie 4 młodsze). Interfejs 4-bitowy jest też opisany w <https://ep.com.pl/files/8691.pdf>

W sprzedaży dostępne są alfanumeryczne wyświetlacze LCD z wieloma innymi metodami komunikacji, np. przez I2C (dwie linie) lub SPI (4 linie).

Sposób definiowania własnych znaków został przedstawiony w tab. 5 na str. 19. Każdy znak tworzy się poprzez wysłanie 8 bajtów do pamięci CGRAM (pierwszy własny znak jest zdefiniowany na ośmiu początkowych bajtach pamięci CGRAM -adresy 0,1,...,7. Drugi znak to kolejne osiem bajtów CGRAM) Gwiazdki oznaczają, że trzy najstarsze bity są nieistotne, gdyż matryca jest 5x8.

W celu zdefiniowania i wyświetleniach dwóch własnych znaków należy wykonać poniższą procedurę:

- 1) Ustawić adres CGRAM = 0
- 2) Zapisać do CGRAM 2\*8 bajtów definiujących kształt 2 własnych znaków (pierwszą instrukcją na str. 26 noty katalogowej)
- 3) Ustawić odpowiedni adres DDRAM określający pozycję na ekranie
- 4) Wyświetlić zdefiniowane znaki poprzez zapisanie do DDRAM wartości 0 (kod pierwszego znaku) i 1 (drugi znak)

Podczas wysyłanie kolejnych wartości do pamięci DDRAM i CGRAM domyślnie następuje automatyczna inkrementacja adresu (nie trzeba po każdym bajcie zapisanym w pamięci wywoływać instrukcji ustawiającej nowy adres)

Wewnętrzny układ resetu zadziała jedynie w sytuacji, gdy są spełnione dość restrykcyjne warunki dotyczące czasy narostu napięcia zasilającego, które są podane na str. 28. Z tego powodu najczęściej wykonuje się inicjalizację programową zgodnie ze schematem przedstawionym na rys. 23-24.

Na stronach 40-44 są pokazane sekwencje instrukcji wyświetlająca napis HITACHI MICROMOC .

### Zadania do wykonania:

Zad. 1 Należy wetknąć wyświetlacz LCD i potencjometr do płytki stykowej oraz wykonać odpowiednie połączenia na płytce stykowej. Linie R/~W proszę podłączyć do masy, gdyż wyświetlacz jest zasilany napięciem 5V i odczytywanie sygnałów z LCD mogłoby spowodować uszkodzenie mikrokontrolera STM32 zasilanego niższym napięciem 3.3V. Należy wykorzystać interfejs 8-bitowy. W celu wygodnego połączenia płytki stykowej z nucleo zalecane jest umieszczenie na płytce stykowej złącza szpilkowego (podwójna listwa goldpin). **UWAGA Proszę uważać, żeby nie podłączyć odwrotnie zasilania ! Do VCC (VDD) powinno być podłączone 5V, a do masa do GND (VSS).**

Zad. 2. Proszę szczegółowo przeanalizować w nocie katalogowej instrukcje z tab. 6 (str. 24-25) oraz procedurę inicjalizacji wyświetlacza pokazaną na rys. 23 na str. 45.

Zad. 3. Należy zaimplementować funkcję inicjalizacji wyświetlacza oraz utworzyć funkcję o nazwie LCD\_Instruction (uint8\_t DB, uint8\_t RS ), która umożliwi wykonanie dowolnego polecenia z tab. 6.

Przykładowo wywołanie LCD\_Instruction (0b01000011, 0) spowoduje, że adres CGRAM będzie 3.

Zad. 4. Proszę poeksperymentować z wywołaniem różnych instrukcji z tab. 6 i zademonstrować ich działanie.

Na ocenę bardzo dobrą konieczne jest dodatkowo wykonanie zad. 5.

Zad. 5. Należy zdefiniować dwa własne znaki i je wyświetlić. Ewentualnie można wykonać prostą animację.

Na ocenę celującą należy również wykonać zad. 6 i 7.

Zad. 6. Należy zaimplementować komunikację za pomocą interfejsu 4-bitowego (bez sprawdzania BUSY FLAG).

### Wskazówki:

Można zmodyfikować wywołanie polecenia Display Off w inicjalizacji pokazanej na rys. 23 na str. 45, ponieważ w nocie katalogowej użyto polecenia Display on/off z parametrami powodującymi wyłączenie wyświetlacza D=0, wyłączenie kursora C=0, wyłączenie miganie kursora B=0. Opcje tego polecenia są pokazane w tab. 6 na str. 24 oraz poniżej:

RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

Znaczenie symboli N, F, D, C, B, I/D, S z rys. 23 jest wyjaśnione pod tab. 6 na str. 25.

Przesuwanie ekranu powinno być wyłączone, natomiast proszę włączyć inkrementacja położenia kursora.

Inicjalizacji wyświetlacza nie można wykonać przed instrukcjami:

```
HAL_Init();
SystemClock_Config();
MX_GPIO_Init();
```

Implementację funkcji LCD\_Instruction można zrealizować za pomocą bezpośrednich operacji na rejestrze ODR odpowiedniego portu np. w przypadku portu C

GPIOC ->ODR=wartość,

ale wówczas uzyskuje się mniej przenośny kod, niż uzyskany za pomocą funkcji z biblioteki HAL.

Uwaga wywołanie GPIOC->ODR wartość może zmienić stan wszystkich szesnastu linii port. W celu modyfikacji tylko 8 najmłodszych bitów należy z GPIOC → ODR najpierw odczytać wartość, a następnie zmodyfikować ją za pomocą odpowiedniej operatorów bitowych & i | np.

```
wartosc = GPIOC->ODR;
wartosc = (wartosc & 0xFF00); //wyzerowanie 8 najmłodszych bitow
wartosc = (wartosc | DB); //ustawienie odpowiednich bitow zgodnie z 8bitowa zmienna DB typu uint8_t
GPIOC->ODR = wartosc;
```

LCD\_Instruction można też utworzyć stosując funkcje z biblioteki HAL w sposób podobny do poniższego:

```
void LCD_Instruction (uint8_t DB, uint8_t RS)
{
    HAL_Delay(2);
    HAL_GPIO_WritePin (LCD_E_GPIO_Port, LCD_E_Pin, SET );

    if (RS)
        HAL_GPIO_WritePin (LCD_RS_GPIO_Port, LCD_RS_Pin, SET );
    else
        HAL_GPIO_WritePin (LCD_RS_GPIO_Port, LCD_RS_Pin, RESET );
    HAL_GPIO_WritePin (LCD_DB7_GPIO_Port, LCD_DB7_Pin, DB & (1<<7) ? SET : RESET);
    HAL_GPIO_WritePin (LCD_DB6_GPIO_Port, LCD_DB6_Pin, DB & (1<<6) ? SET : RESET);
    HAL_GPIO_WritePin (LCD_DB5_GPIO_Port, LCD_DB5_Pin, DB & (1<<5) ? SET : RESET);
    HAL_GPIO_WritePin (LCD_DB4_GPIO_Port, LCD_DB4_Pin, DB & (1<<4) ? SET : RESET);
    HAL_GPIO_WritePin (LCD_DB3_GPIO_Port, LCD_DB3_Pin, DB & (1<<3) ? SET : RESET);
    HAL_GPIO_WritePin (LCD_DB2_GPIO_Port, LCD_DB2_Pin, DB & (1<<2) ? SET : RESET);
    HAL_GPIO_WritePin (LCD_DB1_GPIO_Port, LCD_DB1_Pin, DB & (1<<1) ? SET : RESET);
    HAL_GPIO_WritePin (LCD_DB0_GPIO_Port, LCD_DB0_Pin, DB & (1<<0) ? SET : RESET);
    HAL_Delay(1); //For long RC time constants
    HAL_GPIO_WritePin (LCD_E_GPIO_Port, LCD_E_Pin, RESET );
}
```

Nieodpowiednie ustawienie potencjometru może spowodować, że nic nie będzie widoczne na wyświetlaczu lub wszystkie piksele będą czarne. Należy ustawić kontrast potencjometrem, żeby widoczne były jedynie piksele wyświetlanych znaków.