

Wyświetlacz LCD z układem HD44780

Wyświetlacze alfanumeryczne zamontowane na makietach 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 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
- 2 VCC
- 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.

V0 najczęściej podłącza się do środkowego wyprowadzenia potencjometru wpiętego pomiędzy masę i VCC. (potencjometrem reguluje się, aż będą widoczne piksele). Na makietach potencjometry są już odpowiednio ustawione i nie należy ich przestawiać.

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 może 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 ATMEGI. Druga możliwość polega na użyciu tylko czterech wyprowadzeń DB7-DB4 wraz z RS,RW,E (interfejs 4bitowy - wystarczy 7 linii jednego portu ATMEGI). 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. 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 z <http://www.are.net.pl/?id=products&cmd=9&pid=24> pobrać programy dla ATMEGA32 i rozkompresować wszystkie pliki z katalogu paczka/05-LCD. Podłączyć wyświetlacz do mikrokontrolera zgodnie ze schematem znajdującym się w pliku 5-LCD.jpg. Skompilować i wgrać program lcd.c. **UWAGA:** W makietach taktowanie prawdopodobnie jest ustawione na 8MHz, więc należy odpowiednio zmienić F_CPU w pliku lcd.c (dla za małej wartości czasy opóźnień mogą być zbyt krótkie - https://www.nongnu.org/avr-libc/user-manual/group_util_delay.html)

Zad. 2. Proszę przeanalizować kod z lcd.c. Szczególną uwagę proszę zwrócić na wykorzystanie instrukcji z tab. 6 (str. 24-25) w funkcjach LCD2x16_putchar, LCD2x16_clear i LCD2x16_pos. Należy też porównać kod funkcji LCD2x16_init ze schematem z rys. 23 na str. 45.

Zad. 3. Należy utworzyć funkcję o nazwie instrukcja (unsigned char db, unsigned char rs), która umożliwi wykonanie dowolnego polecenia z tab. 6. Zamiast unsigned char można uint8_t.

Przykładowo wywołanie Instrukcja (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. We wszystkich funkcjach należy zastąpić opóźnienia sprawdzaniem stanu BUSY FLAG (czekać w pętli while do końca wykonywania instrukcji).

Zad. 7. Należy zaimplementować komunikację za pomocą interfejsu 4-bitowego (może być bez sprawdzania BUSY FLAG).