

Uniwersalny szeregowy sterownik wyświetlacza LCD, część 1

AVT-577

Standardowe podłączenie wyświetlacza LCD „zajmuje“ od sześciu nawet do jedenastu wyprowadzeń mikrokontrolera. W przypadku prostego układu nie ma to znaczenia, jeśli jednak w budowanym układzie zaczyna brakować portów warto zastosować wyświetlacz z interfejsem szeregowym.
Rekomendacje: dla twórców systemów wykorzystujących zdalne panele operatorskie, których typowym elementem są wyświetlacze alfanumeryczne.

Pozwoli to na zmniejszenie liczby wyprowadzeń sterujących nawet do jednego, a dodatkowo umożliwi oddalenie wyświetlacza od głównego układu. Takie rozwiązanie zwalnia główny mikrokontroler z konieczności konfiguracji wyświetlacza, gdyż wszystkie procedury wykonuje sterownik wyświetlacza.

Przedstawiony w artykule sterownik wyświetlacza umożliwia sterowanie wyświetlaczem alfanumerycznym o organizacji 2x16 znaków z podświetlaniem poprzez interfejs szeregowy. Przy czym możliwa jest komunikacja w trzech popularnych standardach:

- szeregowy asynchroniczny, zgodny z RS232, z możliwością ustawienia prędkości transmisji w zakresie 2400...5700 bd. W tym trybie wyświetlacz można dołączyć bezpośrednio do mikrokontrolera sterującego lub poprzez prosty tranzystorowy konwerter napięć do złącza szeregowego komputera.
- I²C z możliwością wybrania jednego z czterech adresów układu magistrali, co pozwoli na uniknięcie konfliktów, w przypadku dołączenia kilku układów do magistrali.
- SPI z wykorzystaniem linii wyboru układu (*Slave Select*)- wykorzystywane są trzy linie lub bez wyboru układu - komunikacja odbywa się po dwóch liniach (zegarowej i danych).

W każdej z wymienionych wersji transmisja odbywa się jednokierunkowo (z układu sterującego do wyświetlacza) z maksymalną częstotliwością sygnału zegarowego równą 100 kHz (dla RS232 57600 Hz). Sterownik umożliwia także zdalne sterowanie podświetlaniem wyświetlacza, oprócz jego włączenia lub wyłączenia możliwe jest także ustawienie intensywności podświetlania. Intensywność

może być regulowana w dzieśięciu krokach i jest regulowana poprzez modulację PWM z wykorzystaniem sprzętowego sterownika.

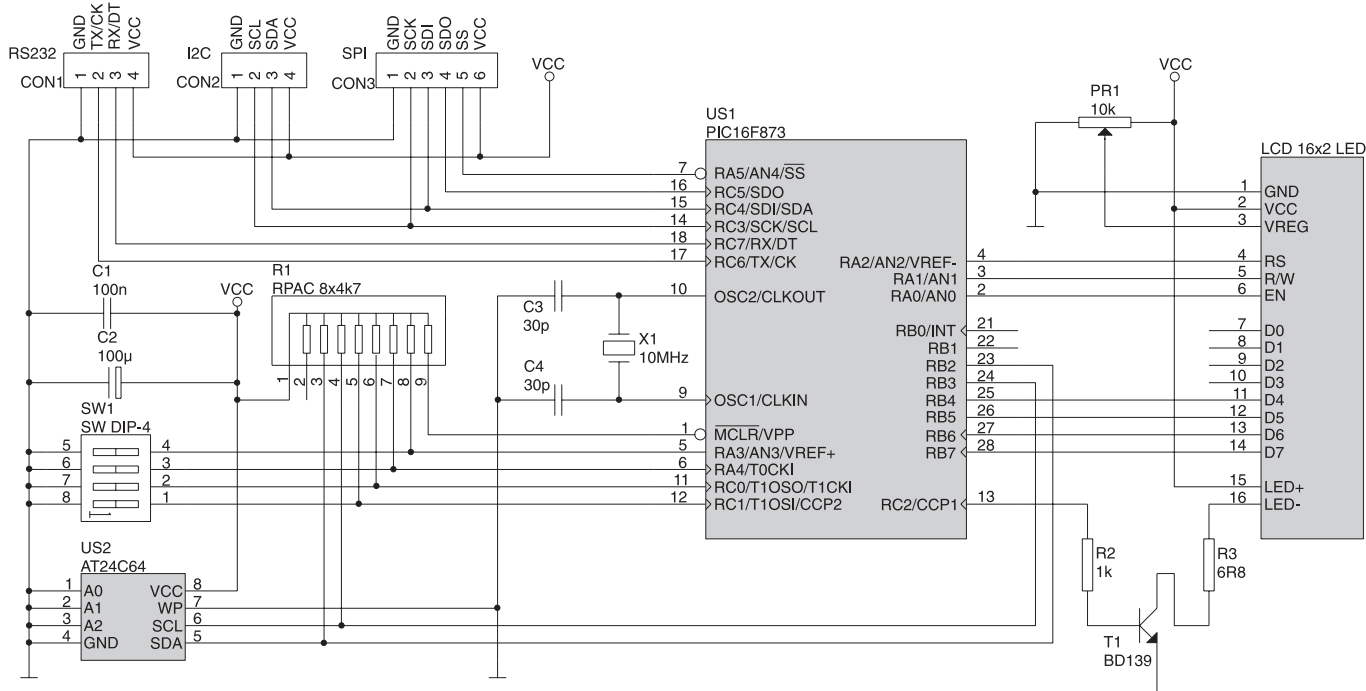
Dodatkowo w układzie sterownika wyświetlacza zastosowana została pamięć EEPROM o pojemności 8 kB służąca do zapisania gotowych komunikatów. Pozwala to na zapisanie w pamięci sterownika maksymalnie 256 gotowych komunikatów, które następnie mogą zostać wyświetlone na wyświetlaczu. Ma to szczególne znaczenie budowie wielopoziomowego menu, gdzie zachodzi potrzeba wyświetlania dużej ilości informacji. Zapisanie ich w pamięci sterownika pozwoli na znaczne zredukowanie potrzebnej pamięci programu mikrokontrolera sterującego. Zapis komunikatów do pamięci wyświetlacza może być wykonane przez interfejs szeregowy lub przy zastosowaniu zewnętrznego programatora pamięci EEPROM.

W pamięci wyświetlacza zapisane są także polskie znaki diakrytyczne, jednak ze względu na ograniczoną pamięć definiowanych znaków mogą być wyświetlane tylko małe znaki z pominięciem „Ż“.

Podstawowe funkcje wyświetlania komunikatów realizowane są przy pomocy znaków ASCII, co w połączeniu z konwerterem RS232->TTL umożliwia obsługę wyświetlacza poprzez dowolny terminal, natomiast zaawansowane funkcje wymagają danych binarnych.

Opis budowy

Schemat elektryczny uniwersalnego sterownika wyświetlacza LCD przedstawiono na **rys. 1**. Elementem sterującym całym układem jest mikrokontroler typu PIC16F873, który pracuje z zewnętrznym rezonatorem kwarcowym



Rys. 1. Schemat elektryczny sterownika wyświetlacza LCD

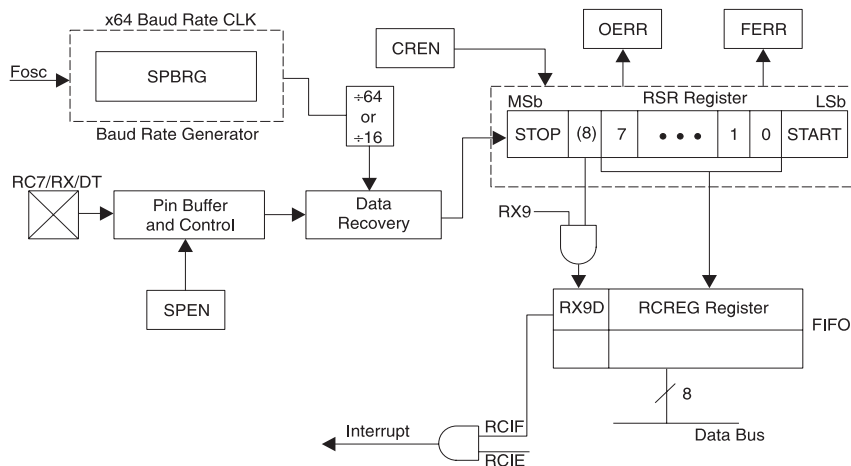
wym X1. Wyświetlacz dołączono do portów RA i RB i pracuje w trybie czterobitowym. Do regulacji kontrastu wyświetlacza służy potencjometr PR1. Jako wzmacniacz prądowy sterujący diodami podświetlającymi wyświetlacz pracuje tranzystor T1, szeregowo włączony rezystor R3 ogranicza maksymalny prąd do wartości około 140 mA. Komunikaty zapisywane są w pamięci EEPROM typu AT24C64 z interfejsem I²C dołączoną do portu RB. Do wyboru sposobu komunikacji sterownika z układem nadrzędnym służy przełącznik SW1. Drabinka rezystorów R1 podciąga wejścia, do których podłączony oraz pamięć EEPROM do plusa zasilania. Linie mikrokontrolera służące do poszczególnych rodzajów transmisji zostały wyprowadzone na złącza odpowiednio: RS232 na złącze CON1, I²C na złącze CON2, SPI na złącze CON3. Na złączu CON1 został wyprowadzony sygnał wyjścia danych RS232 TX/CK jednak nie jest on wykorzystywany ze względu na transmisję jednokierunkową (do wyświetlacza). Również złącze transmisji SPI zawiera sygnał danych wyjściowych SDO, który nie jest wykorzystywany. Dodatkowo, jeśli przy transmisji SPI mniej będzie używane wejście !SS, to układ nadrzędny może zostać dołączony do złącza CON2

przystosowanego do transmisji I²C. W tym przypadku linia SCL jest linią zegarową, a linia SDA linią danych transmisji SPI.

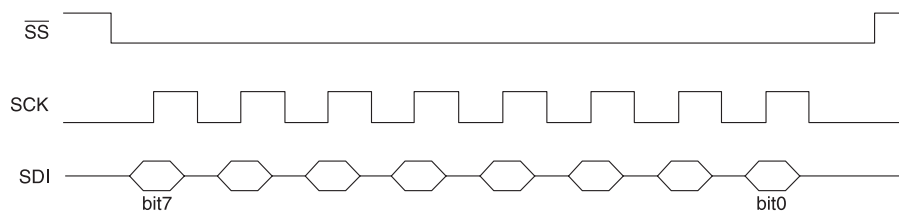
A wszystko sprzętowo

Programowa realizacja obsługi szybkiego interfejsu szeregowego, generowanie sygnału PWM oraz obsługa wyświetlacza wymaga od mikrokontrolera dużej szybkości pracy. Zastosowanie nawet szybkiego mikrokontrolera może i tak nie przynieść pożądanego efektu, gdyż podczas odbioru strumienia danych należy je także analizować i w zależności od wartości poszczególnych bajtów wykonać odpowiednie polecenia i dodatkowo

przesłać je do stosunkowo wolnego wyświetlacza. Aby spełnić wszystkie wymagania czasowe w przestawionym układzie został zastosowany mikrokontroler, który wszystkie operacje związane z transmisją szeregową wykonuje sprzętowo. Dodatkowo posiada sprzętowy sterownik generujący sygnał o zmiennej szerokości impulsu (PWM). Takie rozwiązanie powoduje 8-krotne przyspieszenie jego pracy. Wynika to z faktu, że mikrokontroler nie musi odbierać poszczególnych bitów transmitowanych danych, ale jest informowany o odebraniu całego bajtu przez sterownik sprzętowy. Dzięki temu mikrokontroler ma więcej



Rys. 2. Budowa sprzętowego sterownika transmisji szeregowej USART skonfigurowanego jako odbiornik



Rys. 3. Przebiegi wymagane do poprawnej pracy wyświetlacza w trybie SPI

czasu na analizę danych i obsługę wyświetlacza. Dane są przetwarzane jednakowo dla wszystkich rodzajów interfejsów, a różny jest tylko sposób „składania” bitów w jeden bajt, jednak od strony programowej nie ma to większego znaczenia.

Budowa sprzętowego sterownika dla transmisji RS232 przedstawiono na rys. 2. Po ustawieniu prędkości transmisji poprzez wpis do rejestru SPBRG generowane jest przerwanie gdy odebrany zostanie cały bajt danych. Odebrany bajt znajduje się w rejestrze RCREG.

Odbiór danych w standardzie SPI odbywa się przez moduł MSSP (*Master Synchronous Serial Port*). Ponieważ dane są wysyłane tylko do wyświetlacza wykorzystywane jest tylko wejście danych szeregowych SDI. W takt sygnału zegarowego podawanego przez układ nadrzędny poszczególne bity zapisywane są do rejestru SSPSR, a po odebraniu całego bajtu następuje jego przepisanie do rejestru SSPBUF i wygenerowany zostaje sygnał przerwania,

aby jednostka centralna mikrokontrolera mogła przetworzyć odebrany bajt. W opisywanym układzie sterownik SPI skonfigurowano do pracy z narastającym zboczem sygnału zegarowego, to znaczy stan wejścia SDI jest zapisywany do rejestru SSPSR w momencie zmiany stanu z niskiego na wysoki na wejściu SCK. Dodatkowo istnieje możliwość wyboru, czy dane mają być odbierane przez cały czas, czy dopiero po wybraniu układu (stan niski na wejściu !SS), co umożliwi podłączenie kilku układów z szyną SPI do jednej magistrali. Przebiegi czasowe wymagane do poprawnej pracy są przedstawione na rys. 3. Sygnał !SS jest wymagany tylko przy wyborze takiego trybu pracy. Maksymalna częstotliwość sygnału zegarowego wynosi 100 kHz, ograniczenie to nie wynika z właściwości modułu MSSP, gdyż jest on w stanie odbierać dane z dużo większą prędkością, jednak przy szybszej transmisji mikrokontroler „nie zdąży” ich przetworzyć.

Komunikacja w standardzie I²C odbywa się również przy pomocy

WYKAZ ELEMENTÓW

Rezystory

- R1: RPAC 8*4,7kΩ
- R2: 1kΩ
- R3: 6,8Ω
- PR1: Potencjometr montażowy 10kΩ

Kondensatory

- C1: 100nF
- C2: 100μF/16V
- C3, C4: 30pF

Półprzewodniki

- T1: BD139
- US1: PIC16F873-20MHz zaprogramowany
- US2: AT24C64

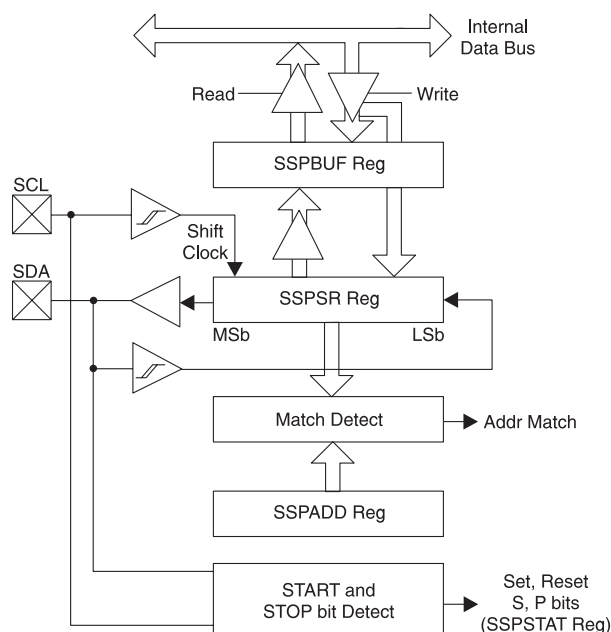
Różne

- CON1, CON2: Goldpin 1x4
- CON3: Goldpin 1x6
- SW1: przełącznik DIP4
- X1: Kwarc 10MHz
- Wyświetlacz LCD 2x16 (GDM1602A)

modułu MSSP, który należy nieco zmodyfikować, schemat blokowy tego modułu przystosowanego do pracy jako I²C Slave jest przedstawiony na rys. 4. Wykorzystywane są także te same rejestry robocze SSPSR i SSPBUF. Praca w trybie Slave jest w pełni kompatybilna ze standardem I²C i dlatego przedstawiony wyświetlacz może zostać dołączony do magistrali wraz z innymi układami, na przykład z pamięcią EEPROM. Adres, pod którym mikrokontroler będzie się zgłaszał na magistrali I²C zależy od wartości wpisanej do rejestru SSPADD. Wszystkie sygnały zgodne ze specyfikacją magistrali I²C są wykrywane i generowane przez sterownik, dzięki czemu jednostka centralna jest „powiadamiana” przerwaniem dopiero w momencie odebrania bajtu danych, co może nastąpić tylko wtedy, gdy adres układu podany na magistrali jest zgodny z zapisanym w rejestrze SSPADD. Jak widać upraszcza to znacznie procedurę obsługi magistrali, a co za tym idzie zajmuje znacznie mniej czasu procesora.

Krzysztof Pławiuk, EP
krzysztof.plawiuk@ep.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: pcb.ep.com.pl oraz na płycie CD-EP5/2004B w katalogu PCB.



Rys. 4. Budowa modułu MSSP w trybie I²C (Slave)

Uniwersalny szeregowy sterownik wyświetlacza LCD, część 2

AVT-577



W drugiej części artykułu przedstawiamy sposób wykonania oraz obsługę i konfigurację modułu wyświetlacza.

Rekomendacje: dla twórców systemów wykorzystujących zdalne panele operatorskie, których typowym elementem są wyświetlacze alfanumeryczne.

Kolejnym modulem przyspieszającym pracę mikrokontrolera jest sprzętowy sterownik PWM, który został wykorzystany do regulacji intensywności podświetlania wyświetlacza. Do sterowania podświetlaniem wyświetlacza wykorzystano wyprowadzenie RC2 procesora. Wyprowadzenie to oprócz funkcji typowego wyprowadzenia wejścia/wyjścia jest także wyjściem sygnału sprzętowego sterownika PWM, dzięki temu jest możliwe elektroniczne ustawienie intensywności podświetlania wyświetlacza.

Sprzętowy sterownik PWM sprawia, że sterowanie podświetlaniem wyświetlacza jest wykonywane niezależnie od pracy jednostki centralnej procesora. Do pracy sterownika wykorzystywany jest licznik TMR2, który wraz z komparatorami cyfrowymi umożliwia wytworzenie na wyjściu RC2 przebiegu o dowolnym wypełnieniu bez udziału jednostki centralnej procesora.

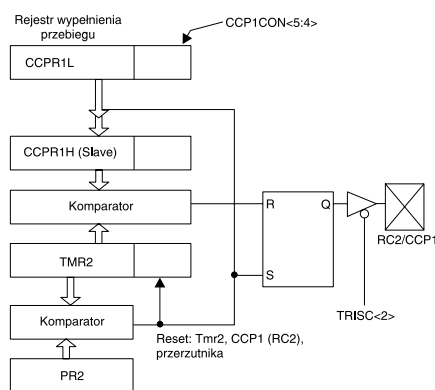
Konfigurację licznika TMR2 w trybie sprzętowego sterownika PWM przedstawiono na rys. 5. Generowanie przebiegu o zmiennym wypełnieniu sprowadza się do wpisania do rejestru PR2 czasu trwania jednego cyklu, a do

rejestru CCPR1L czasu trwania stanu wysokiego na wyjściu RC2. Po wpisaniu tych wartości przebieg na wyjściu RC2 jest wytwarzany przez TMR2.

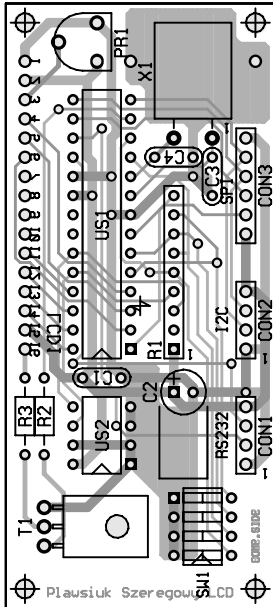
Pomimo przeniesienia wszystkich funkcji transmisji danych na sterowniki sprzętowe, podczas odbioru strumienia danych występuje problem zbyt wolnego wykonywania poleceń przez wyświetlacz, gdyż na przykład kasowanie całego wyświetlacza trwa około 2 ms. Wysyłane dane do wyświetlacza w tym czasie zostałyby utracone. Aby zapobiec takiej sytuacji, w pamięci mikrokontrolera została utworzona 64-bajtowa pamięć FIFO. Odbierane dane z układu sterującego kierowane są do tej pamięci, tak aby żaden bajt nie został „zgubiony”, a następnie poszczególne bajty są analizowane i w zależności od ich wartości wykonywane są odpowiednie działania (kierowanie do wyświetlacza, zmiana intensywności podświetlania).

Montaż i uruchomienie

Moduł sterujący został zmontowany na płytce dwustronnej (schemat montażowy pokazano na rys. 6) wielkością dopasowaną do rozmiarów wyświetlacza. Montaż należy rozpocząć od elementów o najmniejszych gabarytach. Tranzystor T1 i rezonator kwarcowy montowane są na leżąco. Złącza CON1, CON2 i CON3 są montowane od strony elementów, natomiast wyświetlacz od strony lutowania. Po zamontowaniu wszystkich elementów można przejść do uruchomienia układu. W tym celu do dowolnego złącza CON1...CON3 należy dołączyć napięcie zasilania o wartości 5 V, zwracając szczególną uwagę na odpowiednią polaryzację, gdyż układ nie jest zabezpieczony przed odwrotną polaryzacją. Zasilacz powi-



Rys. 5. Budowa sprzętowego sterownika PWM



Rys. 6. Schemat montażowy płytki drukowanej

nien mieć wydajność prądową co najmniej 150 mA (w przypadku wyświetlacza z podświetlaniem) i 10 mA dla wyświetlacza bez podświetlania. Zastosowany wyświetlacz jest skonfigurowany tak, aby podświetlanie działało cały czas, po podłączeniu napięcia. Aby umożliwić sterowanie poprzez mikrokontroler, należy na płycie wyświetlacza zdemontować rezystory oznaczone jako R6 i R7. Na rys. 7 przedstawiono schemat prostego konwertera napięć, umożliwiającego podłączenie wy-

Tab. 1. Stan przełącznika SW1 i odpowiadające mu tryby pracy wyświetlacza

RS232 S1=0, S2=0		
S3	S4	Prędkość
0	1	2400b
1	0	9600b
0	1	19200b
1	1	57600b
I2C S1=1, S2=0		
S3	S4	Adres
0	0	90h
1	0	92h
0	1	94h
1	1	96h
SPI S1=0, S2=1 -!SS		
S3	S4	
x	x	
SPI S1=1, S2=1		
S3	S4	
x	x	

świetlacza bezpośrednio do portu szeregowego komputera i wstępne sprawdzenie działania wyświetlacza bez konieczności oprogramowania mikrokontrolera sterującego.

Obsługa

Rodzaj interfejsu jest wybierany za pomocą przełącznika SW1. Dostępne są cztery tryby pracy interfejsu szeregowego: RS232, I²C, SPI oraz SPI z sygnałem wyboru układu !SS. Stan przełącznika oraz odpowiadające mu tryby pracy przedstawiono w tab. 1. Rodzaj komunikacji jest wybierany za pomocą przełączników S1 i S2, przełączniki S3 i S4 natomiast służą do zmiany parametrów danego typu komunikacji. Wszystkie przełączniki wchodzą w skład przełącznika SW1, stan 0 odpowiada ustawieniu przełącznika w pozycję ON (wejście mikrokontrolera jest zwierane do masy). Dla pracy w trybie RS232 można wybrać w zależności od potrzeb jedną z czterech prędkości pracy. Dla pracy w trybie I²C przełączniki S3, S4 umożliwiają zmianę adresu, pod którym będzie się zgłaszał wyświetlacz na magistrali I²C. Ta możliwość pozwala uniknąć konfliktów w przypadku podłączenia innego układu o takim samym adresie lub dwóch prezentowanych wyświetlaczy.

Trzeci tryb odnosi się do pracy w trybie SPI z aktywnym wejściem !SS, natomiast w czwartym trybie do pracy nie jest wymagany sygnał !SS. Dla komunikacji w trybie SPI przełączniki S3 i S4 nie są używane, dlatego ich stan nie ma znaczenia.

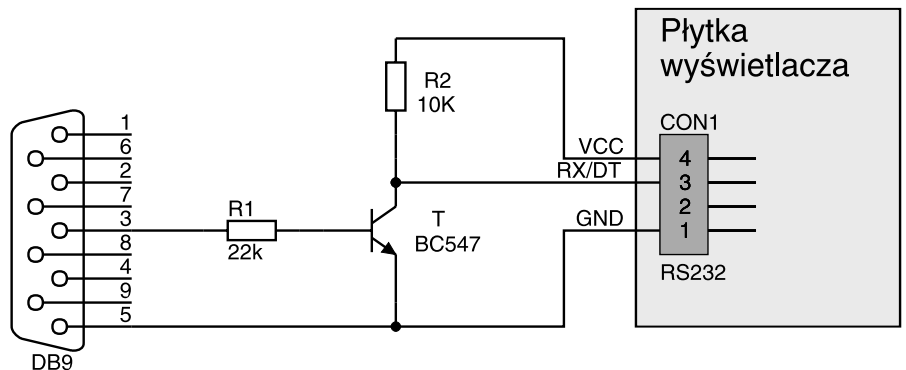
Obsługa wyświetlacza jest jednakowa niezależnie od użytego interfejsu. W podstawowej formie

Tab. 2. Wartości hex przypisane dla polskich znaków diakrytycznych

Znak	Wartość hex	Kombinacja klawiszy
Ą	0xB9	Alt+a
Ć	0xE6	Alt+c
Ę	0xEA	Alt+e
Ł	0xB3	Alt+l
Ń	0xF1	Alt+n
Ś	0x9C	Alt+s
Ó	0xF3	Alt+o
Ż	0xBF	Alt+z

do wyświetlacza mogą być wysyłane znaki ASCII, które będą wyświetlane na wyświetlaczu. Dostęp do pamięci EEPROM czy sterowania podświetlaniem wymaga użycia komend spoza znaków ASCII. Kody polskich znaków diakrytycznych są zgodne z kodami generowanymi przy wpisywaniu tych znaków poprzez standardową klawiaturę komputera, dlatego po połączeniu wyświetlacza z komputerem można je wpisywać w analogiczny sposób (ALT+znak). Ze względu na ograniczoną ilość pamięci znaków wyświetlacza możliwe jest wyświetlenie tylko ośmiu znaków. Spis wszystkich znaków oraz odpowiadające im wartości podano w tab. 2. W przypadku sterowania wyświetlaczem przez na przykład mikrokontroler, aby wyświetlić dany znak, należy wysłać jego wartość szesnastkową.

Oprócz kodów znaków typowych dla wyświetlania informacji, wyświetlacz rozpoznaje dodatkowo siedem kodów służących do wydawania poleceń. Spis wszystkich kodów znajduje się w tab. 3.



Rys. 7. Schemat odłączenia wyświetlacza z komputerem poprzez prosty konwerter poziomów.

Tab. 3. Wartości hex służące do konfiguracji wyświetlacza

Wartość (hex)	Funkcja
0x1B	Kasuje cały wyświetlacz i ustawia kursor na pierwszej pozycji
0x01	Ustawia kursor na pierwszej pozycji pierwszej linii
0x02	Ustawia kursor na pierwszej pozycji drugiej linii
0x03	Umożliwia ustawienie kursora na podanej pozycji 1...32
0x04	Umożliwia ustawienie intensywności podświetlania
0x05	Umożliwia zapis komunikatów do pamięci EEPROM
0x06	Umożliwia odczyt komunikatów z Eeprom i wyświetlanie ich na wyświetlaczu

Pierwszy, o wartości 0Bh, służy do kasowania całego wyświetlacza, kod ten posiada taką samą wartość jak naciśnięcie klawisza ESC na klawiaturze komputera. Wartości 01h i 02h służą do ustalenia pozycji kursora na pierwszej pozycji odpowiednio linii pierwszej lub drugiej bez jej kasowania. Podanie wartości 03h umożliwia natomiast ustawienie kursora na dowolnej pozycji wyświetlacza. Komenda ta składa się z dwóch bajtów: rodzaju komendy (03h) oraz pozycji kursora (1...32d), z tym że wartość 1...16 ustawia kursor w linii pierwszej, natomiast wartość z przedziału 17...32 ustawia kursor w linii drugiej. Przykładowe polecenie ustawiające kursor na pozycji numer 2 w linii drugiej będzie miało postać „03h18d”.

Komenda o wartości 04h służy do ustawienia intensywności podświetlania wyświetlacza, podobnie jak poprzednio składa się ona z dwóch bajtów: sygnatury komendy (04h) oraz wartości parametru. Wartość parametru może być zmieniana w zakresie 00h...09h, parametr też może być również podany jako znak ASCII w zakresie 0...9. Regulacja intensywności może być zmieniana w dziesięciu krokach, z tym że wartość zero wyłącza całkowicie podświetlanie, a wartość 9 włącza z maksymalną intensywnością.

Komendy o wartościach 05h i 06h odnoszą się do komunikatów umieszczonych w pamięci EEPROM: 05h dotyczy zapisu komunikatów do pamięci, natomiast 06h dotyczy ich odczytu i wyświetlenia na wyświetlaczu. Wydane polecenie w obydwu przypadkach składa się z dwóch bajtów: sygnatury 05h lub 06h oraz numeru komunikatu o wartości z zakresu 00h...FFh (0...255 dziesiętnie). Na podstawie numeru

komunikatu obliczany jest adres, pod którym zostanie zapisany w pamięci EEPROM. Ponieważ każdy komunikat składa się z 32 znaków, adres ten wynika z zależności: adres = (numer komunikatu) * 32 (tę zależność należy także wykorzystać w przypadku zapisu komunikatów poprzez zewnętrzny programator pamięci EEPROM). Przy zapisie poprzez interfejs szeregowy sterownika wyświetlacza dane komunikatu, który ma być zapisany w pamięci EEPROM, pobierane są z pamięci wyświetlacza, tak więc wydanie komendy zapisu powoduje zapis aktualnie wyświetlanych danych na wyświetlaczu. Aby zapisać komunikat w pamięci, należy go wcześniej wyświetlić na wyświetlaczu, a następnie wydać polecenie zapisu do pamięci, na przykład (0500h) spowoduje zapis stanu wyświetlacza do pamięci jako komunikat numer 0. Po zapisaniu każdego komunikatu należy odczekać minimum 10 ms, aby został zapisany w pamięci. Czas ten jest wymagany do zapisu jednej komórki pamięci EEPROM, jednak zastosowana pamięć posiada możliwość zapisu stronicowego, każda strona może składać się z maksymalnie 32 bajtów, dlatego jednorazowo jest zapisywana cała zawartość wyświetlacza. Wyświetlenie komunikatów zawartych w pamięci wykonuje się w sposób analogiczny jak zapis, ale jako komendę należy podać wartość 06h, a następnie numer komunikatu. Komenda wyświetlająca komunikat numer 0 będzie miała postać 0600h.

Krzysztof Pławiuk, EP
krzysztof.plawsiuk@ep.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: pcb.ep.com.pl oraz na płycie CD-EP6/2004B w katalogu PCB.