

kit

2988

AVT

Wyświetlacz tekstowy LED

Wyświetlacz tekstowy LED służy do wyświetlania przesuwanego się tekstu. Zastosowane nowoczesne diody LED zapewniają widoczność napisu na zewnątrz budynku, także w słoneczne dni. Do wprowadzania tekstu do wyświetlacza jest wykorzystywana standardowa klawiatura komputerowa (PS/2). Urządzenie posiada wewnętrzną pamięć nieulotną, która nie kasuje się po odłączeniu zasilania. Wyświetlacz składa się z sześciu matryc 8 x 8 z diod czerwonych LED o światłości 4000mcd.

Prezentowany układ charakteryzuje się łatwą obsługą, a tekst (do 400 znaków) może wyświetlać ze zmienną prędkością.

Celem budowy tego wyświetlacza była chęć autoprezentacji własnej osoby podczas szuka-

nia pracy. Jednak udało mi się znaleźć pracę znacznie szybciej, niż sądziłem.

Opis układu

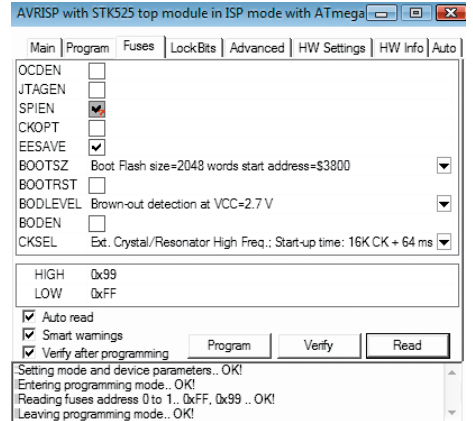
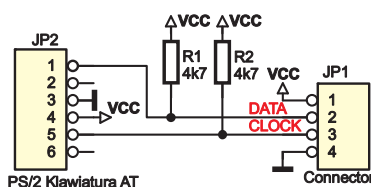
Urządzenie zawiera:

- sterownik (rysunek 1),
- sześć modułów matrycy 8 x 8 (rysunek 2),
- złącze klawiatury PS/2 (rysunek 3).

Sercem jest popularny mikrokontroler Atmel ATmega16, który ma 32 linie I/O, wyprowadzone na listwy szpilkowe JP5... JP12. Urządzenie zasilane jest przez transformator TS10/35 Złącza goldpin JP13, JP14 ułatwiają rozbudowę urządzenia. W urządzeniu zastosowano rezonator kwarcowy 8MHz i kondensatory C5, C6 (22 pF)

Ustawienie bitów konfiguracyjnych (ang. *fuse bits*) pokazano na rysunku 4.

Rys. 3 Złącze klawiatury



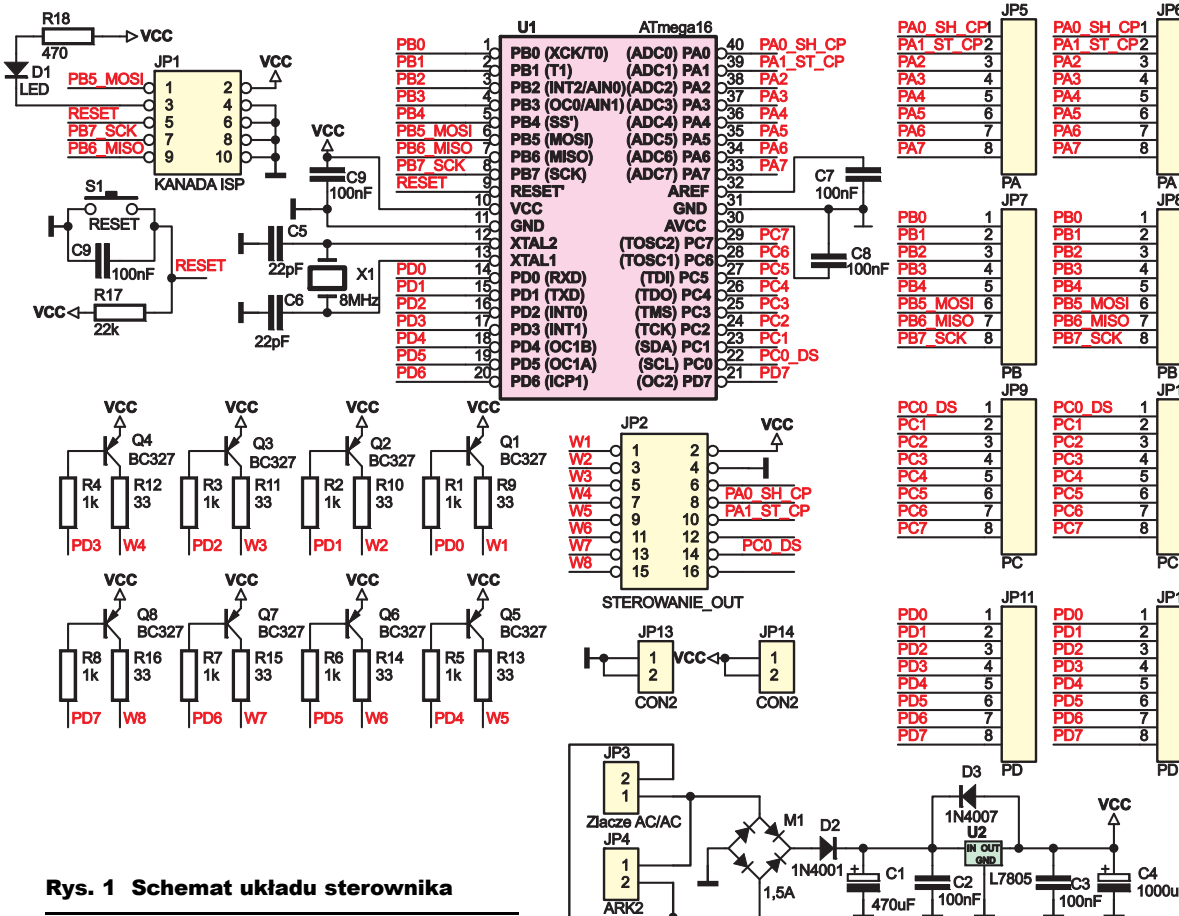
Rys. 4 Fuse bits

Mikrokontrolery AVR zawierają interfejs ISP (ang. *In-System Programmable*) – złącze JP1 służy do programowania mikrokontrolera.

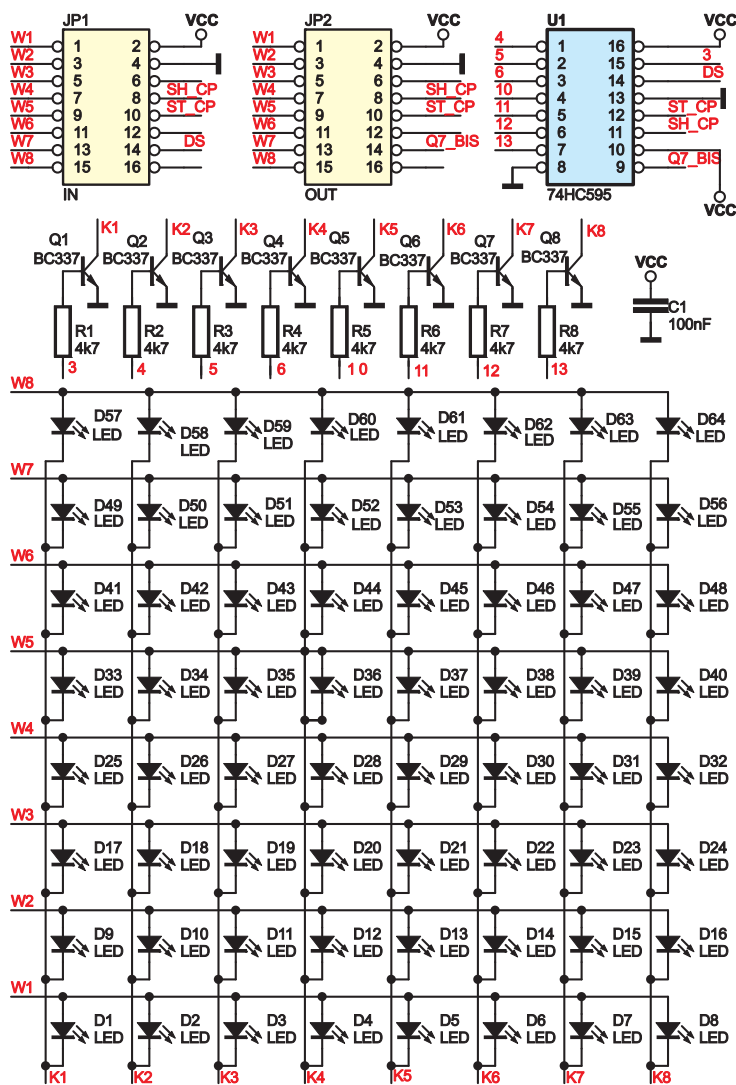
Wyświetlacz jest matrycowy, multiplek-

sowany. Poszczególne kolumny zaświecają się tylko na chwilę. O zaświeceniu diody w odpowiednim wierszu decyduje stan niski na wyjściach PD0-PD7 (PORTD). Stan niski otwiera tranzystory PNP BC327 Q1-Q8 przez rezystory o R1-R8 (1kΩ). Nasylenie odpowiednich tranzystorów Q1-Q8 powoduje podanie na odpowiednie wiersze (W1...W8) dodatniego napięcia przez rezystory R9-R16 (33Ω), ograniczające prąd do około 100mA.

Kolumny sterowane są w poszczególnych sześciu matrycach za pomocą rejestrów – buforów 74HC595, sterowanych szeregowo tylko trzema sygnałami DS, SH CP, ST CP. Sygnały sterujące wierszami W1-W8 i rejestrem



Rys. 1 Schemat układu sterownika



Rys. 2 Schemat modułu matrycy 8x8

przesuwnym 74HC595 oraz napięcie zasilania zostały doprowadzone do złącza JP2, które łączy układ sterownika z pierwszym układem modułu matrycy 8 x 8, a potem z kolejnymi. Połączenia należy dokonać za pomocą taśmy 16-żyłowej. W opisywanej wersji złącza JP5... JP12 nie są wykorzystane.

Jako interfejs komunikacyjny zastosowano klawiaturę PC AT. Rysunek 3 przedstawia przejściówkę klawiatury. Sygnał DATA należy podłączyć przewodem do nóżki PB0, a sygnał CLOCK do nóżki PB2 (przerwanie INT2). Rezystory zwiększają odporność na zakłócenia. Rysunek 5 przedstawia wyprowadzenie złącza klawiatury PC w standardzie AT.

Sercem modułu matrycy 8 x 8 (rysunek 2) jest szeregowy rejestr przesuwany 74HC595 z szeregowym wejściem i 8-bitowym równoległym wyjściem. Informacje z wejścia danych DS są przesuwane w rejestrze w kolejnych

cyklach sygnału zegarowego SH_CP. ST_CP to sygnał powodujący przeniesienie danych z rejestru do zatrasku i na wyjście układu. Połączenie anod diod LED tworzy wiersze W1-W8 matrycy, natomiast połączenie katod diod tworzy kolumny K1-K8. Wykorzystano diody czerwone LED LL-503 VC2E-V1-2BC firmy LUCKY LIGHT, 5mm, o światłości 4000 mcd przy prądzie $I_F=20\text{mA}$ i napięciu przewodzenia typowo 2,05V. Chwilowy prąd przewodzenia może wynieść 100mA. Jedyną wadą to dosyć wąski kąt świecenia diody (20°) [7]. Diody to najdroższa część wyświetlacza; można zastosować tańsze, byle odpowiednio wydajne. Jeżeli wyświetlacz nie będzie montowany w wąskim korytarzu lub ulicy, to należałoby zaopatrzyć się w diody o większym kącie świecenia. Z a ś w i e c a n i e kolumn zostało zrealizowane za pomocą tranzystorów Q1-Q8, sterowanych z rejestru 74HC595 poprzez rezystory R1-R8. Bezpośrednie dołączenie kolumn matrycy do wyjść układu scalonego 74HC595 jest niemożliwe ze względu na wymagany prąd, maksymalnie 800mA. Tranzystory powinny mieć duże wzmocnienie, by nie pojawiły się różnice w jasności świecenia poszczególnych kolumn, zmieniające się wraz z liczbą zapalonych diod. Złącze JP3 służy do połączenia ze złączem JP2 sterownika lub ze złączem JP4 poprzedniego modułu matrycy 8 x 8. Połączenia należy dokonać za pomocą taśmy 16-żyłowej.

Zasada sterowania

Mikrokontroler odczytuje z pamięci wzór pierwszej kolumny dla wyświetlanego znaku i ustawia PORTD, który steruje tranzystorami kolumn BC327. Tranzystory podają napięcie dodatnie na anody

odpowiednich (w zależności od wyświetlanego znaku) diod. Jednocześnie mikrokontroler podaje sygnały sterujące dla rejestru przesuwającego przeniesienie danych z rejestru do zatrasku i na wyjście układu. Połączenie anod diod LED tworzy wiersze W1-W8 matrycy, natomiast połączenie katod diod tworzy kolumny K1-K8. Wykorzystano diody czerwone LED LL-503 VC2E-V1-2BC firmy LUCKY LIGHT, 5mm, o światłości 4000 mcd przy prądzie $I_F=20\text{mA}$ i napięciu przewodzenia typowo 2,05V. Chwilowy prąd przewodzenia może wynieść 100mA. Jedyną wadą to dosyć wąski kąt świecenia diody (20°) [7]. Diody to najdroższa część wyświetlacza; można zastosować tańsze, byle odpowiednio wydajne. Jeżeli wyświetlacz nie będzie montowany w wąskim korytarzu lub ulicy, to należałoby zaopatrzyć się w diody o większym kącie świecenia. Z a ś w i e c a n i e kolumn zostało zrealizowane za pomocą tranzystorów Q1-Q8, sterowanych z rejestru 74HC595 poprzez rezystory R1-R8. Bezpośrednie dołączenie kolumn matrycy do wyjść układu scalonego 74HC595 jest niemożliwe ze względu na wymagany prąd, maksymalnie 800mA. Tranzystory powinny mieć duże wzmocnienie, by nie pojawiły się różnice w jasności świecenia poszczególnych kolumn, zmieniające się wraz z liczbą zapalonych diod. Złącze JP3 służy do połączenia ze złączem JP2 sterownika lub ze złączem JP4 poprzedniego modułu matrycy 8 x 8. Połączenia należy dokonać za pomocą taśmy 16-żyłowej.

odpowiednich (w zależności od wyświetlanego znaku) diod. Jednocześnie mikrokontroler podaje sygnały sterujące dla rejestru przesuwającego przeniesienie danych z rejestru do zatrasku i na wyjście układu. Połączenie anod diod LED tworzy wiersze W1-W8 matrycy, natomiast połączenie katod diod tworzy kolumny K1-K8. Wykorzystano diody czerwone LED LL-503 VC2E-V1-2BC firmy LUCKY LIGHT, 5mm, o światłości 4000 mcd przy prądzie $I_F=20\text{mA}$ i napięciu przewodzenia typowo 2,05V. Chwilowy prąd przewodzenia może wynieść 100mA. Jedyną wadą to dosyć wąski kąt świecenia diody (20°) [7]. Diody to najdroższa część wyświetlacza; można zastosować tańsze, byle odpowiednio wydajne. Jeżeli wyświetlacz nie będzie montowany w wąskim korytarzu lub ulicy, to należałoby zaopatrzyć się w diody o większym kącie świecenia. Z a ś w i e c a n i e kolumn zostało zrealizowane za pomocą tranzystorów Q1-Q8, sterowanych z rejestru 74HC595 poprzez rezystory R1-R8. Bezpośrednie dołączenie kolumn matrycy do wyjść układu scalonego 74HC595 jest niemożliwe ze względu na wymagany prąd, maksymalnie 800mA. Tranzystory powinny mieć duże wzmocnienie, by nie pojawiły się różnice w jasności świecenia poszczególnych kolumn, zmieniające się wraz z liczbą zapalonych diod. Złącze JP3 służy do połączenia ze złączem JP2 sterownika lub ze złączem JP4 poprzedniego modułu matrycy 8 x 8. Połączenia należy dokonać za pomocą taśmy 16-żyłowej.

Program

Kod programu został napisany w języku C i skompilowany w środowisku AVR Studio z WinAVR. Program po skompilowaniu zajmuje około 43% pamięci Flash (7KB). Cały program składa się z kilku plików (można je ściągnąć z Elportalu).

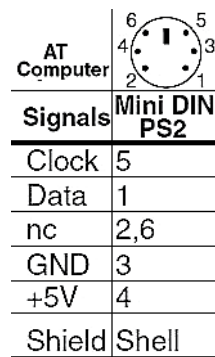
W programie każdy plik źródłowy ma odpowiadający mu plik nagłówkowy *.h, gdzie umieszczone są deklaracje funkcji oraz zmiennych zewnętrznych. Oto ważniejsze funkcje:

- *initKeyboard()* – ustawienie przerwania INT2, które jest wywoływane po wykryciu zbocza opadającego na nóżce PB2 i jest wykorzystane do odebrania i zamiany kodu klawisza klawiatury AT na wartość dziesiętną znak ASCII.

- *initDisplay()* – inicjalizuje TIMER1 w tryb CTC, polegający na zerowaniu licznika po wykryciu zgodności odpowiednich rejestrów zliczających mikrokontrolera i rozpoczęciu zliczania od początku. TIMER1 skonfigurowano tak, że zerowanie licznika powoduje wywołanie funkcji obsługi przerwania od TIMER1 (*ISR(TIMER1_COMPA_vect)*).

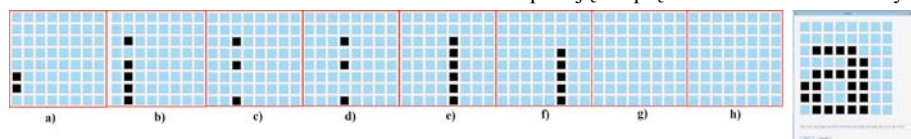
Przerwanie wykorzystuje się do generowania programowej magistrali szeregowo SPI oraz przesuwania napisu na wyświetlaczu lub menu konfiguracyjnego.

Następnie pętla *for* odczytuje wartości dziesiętne znaków ASCII z pamięci EEPROM, które będą prezentowane na wyświetlaczu. Znaki są zapisywane do tablicy *tekst[]*. Znak ASCII jest

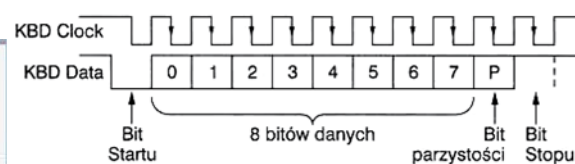


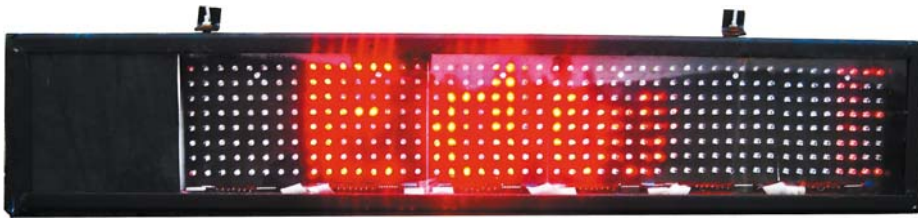
Rys. 5 Piny gniazda PS2

Rys. 6 Idea wyświetlania znaku



Rys. 7 Transmisja danych z klawiatury





reprezentowany przez wartość dziesiętną znaku przesuniętą o 31 w dół (np. odstęp (spacja) z wartości 32 staje się 1). ZERO (0) jest informacją o końcu tablicy znaków (podobnie jak w łańcuchach znaków). W dalszej części programu jest wywoływana funkcja *sei()*, która powoduje globalne włączenie przerwań. Ostatnim krokiem w funkcji *main()* jest przejście do pętli nieskończonej *while(1)*, w której realizuje się główne zadania umożliwiające poprawną pracę urządzenia. Pętla *while(1)* sprawdza, czy nie ma nowych znaków ASCII do zapisu w tablicy *tekst[]*. Jeżeli odebrano nowy znak, na końcu tablicy *tekst[]* jest dopisywana pionowa belka (znak zachęty |), następnie jest wywoływana funkcja *srodkowanie()*. Realizuje ona wyświetlenie ostatnich znaków z tablicy *tekst[]* na wyświetlaczu.

Znak ASCII, który znajduje się w zmiennej *abc*, jest dopisywany do tablicy *tekst[]*, zwiększana jest zmienna *ilosc_liter*, dopisywana pionowa belka i ponownie wywoływana funkcja *srodkowanie()*. Sytuacja się powtarza do momentu osiągnięcia przez zmienną *ilosc_liter* maksymalnej liczby możliwych liter do zapisania lub zakończenia wprowadzania znaków poprzez wciśnięcie klawisza „Enter” na klawiaturze AT (zmienna *zapis_eeprom* przyjmuje wartość JEDEN). Wówczas pionowa belka jest nadpisywana liczbą 0, co jest informacją o końcu tablicy *tekst[]*. Wprowadzone znaki są zapisywane do pamięci EEPROM (zapis jest poprzedzony globalnym wyłączeniem przerwań – funkcja *cli()*). Po zakończonym zapisie ponownie zostają włączone przerwania globalne poprzez funkcję *sei()*.

Usuwanie znaków (klawisz Backspace) polega na nadpisaniu znaku znakiem spacji, dopisaniu znaku pionowej belki i wywołaniu funkcji *srodkowanie()*.

Łatwiejszy w implementacji programowej jest przypadek transmisji danych z klawiatury do mikroprocesora. Przebiegi w takim przypadku przedstawia **rysunek 7**.

Przesyłane słowa składają się z 11 bitów: bit startu (zawsze równy ZERO), osiem bitów danych, począwszy od najmniej znaczącego, bit parzystości (typu „Odd” – dopełnienie do

parzystej liczby jedynek w transmitowanym bajcie) oraz bit stopu, zawsze równy JEDEN. Odczyt linii danych następuje przy opadającym zboczku sygnału CLOCK. Prędkość transmisji nie jest zbyt duża i nie podlega żadnej dokładnej standaryzacji. Częstotliwość sygnału zegarowego zazwyczaj zawiera się w przedziale 20...30kHz. Współpracujący mikrokontroler może sterować transmisją z klawiatury przez wymuszenie poziomu niskiego na linii CLOCK. Niski poziom oznacza blokadę transmisji. Rozpoczęcie nadawania danych nastąpi dopiero po pojawieniu się na tej linii poziomu wysokiego.

Każda zmiana stanu klawisza powoduje przesłanie 8-bitowego kodu klawisza. Kody odpowiadające poszczególnym klawiszom klawiatury przedstawiono na **rysunku 8**. Jeżeli klawisz jest przytrzymany, po pewnym opóźnieniu transmisja jest powtarzana z określoną częstotliwością. Po zwolnieniu klawisza klawiatura przesyła wartość 0xF0 (zwiastun kodu klawisza puszczonego). Chwilę po nim przesyłany jest kod klawisza, który został puszczonego. Przykładowo, naciśnięcie i puszczenie klawisza „Z” powoduje transmisję następującego ciągu wartości: 0x1A, 0xF0, 0x1A. Niestety wartości przyporządkowane poszczególnym klawiszom nie są kodami ASCII. Przepisano je „na sztywno” i nie zależą od stanu klawisza *Shift*. Klawisze *Shift*, *Alt*, *Ctrl* są traktowane przez klawiaturę tak samo, jak klawisze znakowe. Niektóre z kodów klawiszy są dwu- i więcej bajtowe – tzw. klawisze specjalne. Naciśnięcie jednego z nich powoduje przesłanie zwiastuna klawisza specjalnego (kod 0xE0) przed właściwą jego wartością. W przypadku puszczenia, w pierwszej kolejności przesyłany jest zwiastun specjalny, a dopiero później zwiastun klawisza puszczonego (przykładowo, puszczenie klawisza Alt powoduje przesłanie następującego ciągu znaków: 0xE0, 0xF0, 0x11) [1].

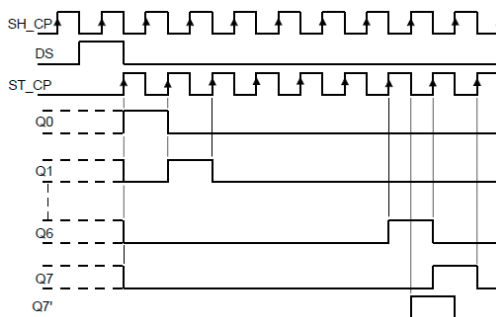
Program odbioru danych z klawiatury został napisany jako obsługa przerwania zewnętrznego (*ISR(INT2_vect)*) aktywowanego zboczem opadającym. Kod obsługi przerwania odczytuje kolejne bity pojawiające się na linii DATA w takt sygnału zegarowego CLOCK doprowadzonego do linii PB2. Opadające zbocze syg-

nału zegarowego wywołuje procedurę obsługi przerwania, odczytującą kolejno wszystkie transmitowane bity przez klawiaturę (rysunek 7). Przy odbiorze dziewiątego bitu sprawdzana jest parzystość odebranych danych i porównywana z bitem parzystości. W przypadku przekłamania dane są ignorowane. Prawidłowo odebrana dana jest dekodowana. Pierwszym klawiszem, który można jedynie wcisnąć (inne są ignorowane) w celu konfiguracji wyświetlacza lub tekstu, jest Esc. Wówczas zostaje ustawiony odpowiedni status (zmienna *status*) oraz ustawiona zmienna *menu_glowne* w celu wyświetlenia menu konfiguracyjnego (1 – Tekst, 2 – Prędkość, 3 – Czas/Data/Temp., 4 – Opcje). Wciśnięcie klawiszy 1...4 (inne są ignorowane) powoduje przypisanie zmiennej *menu_glowne* nowej wartości i wyświetlenie kolejnego menu konfiguracyjnego (szczegółowy opis w Obsłudze urządzenia). Wybór 1 przenosi menu do możliwości edycji tekstu. Odebrany kod klawisza przechowywany w zmiennej *kod_klawisza* jest dekodowany na kod ASCII za pomocą pętli *for*. Pętla *for* za pomocą funkcji *sign_normal()* szuka kodu ASCII w pliku *scancode.h* (tablica z kodami klawiszy klawiatury ułożona zgodnie z kolejnością tablicy ASCII), który odpowiada kodowi odebranego klawisza. Jeżeli pętla *for* odnajdzie odebrany kod klawisza, wówczas licznik pętli *for*, jako kod ASCII, jest przepisywany do zmiennej *abc*, której wartość jest zapisywana w tablicy *tekst[]* w pętli głównej programu (*while(1)*). W przypadku odebrania błędnych danych, wysyłany jest odpowiednio określony kod, który informuje pętlę główną, że dane są błędne. Jeżeli wciśnięto wcześniej klawisz Caps Lock, to kod ASCII jest szukany za pomocą funkcji *sign_caps()*, a jeżeli wciśnięto prawy Alt, to wykorzystywana jest funkcja *sign_polskie_alt()*. Oczywiście wciśnięcie danego klawisza specjalnego powoduje odpowiednie ustawienie zmiennej *status*. Takie rozwiązanie eliminuje traktowanie przez klawiaturę wszystkich klawiszy jednakowo. Wciśnięcie klawisza Backspace powoduje wysłanie określonego kodu, który w programie głównym jest interpretowany jako instrukcja usunięcia ostatniego znaku z tablicy *tekst[]*. Potwierdzenie zakończenia wprowadzania danych dokonuje się naciśnięciem klawisza Enter. Wówczas są zerowane zmienne odpowiadające za przesuwanie tekstu, ustawienie zmiennej *zapis_eeprom* oraz *menu_glowne*. Na wyświetlaczu można zaobserwować (już po skończonym zapisie danych do pamięci EEPROM) ostatnio dokonane zmiany w tekście. Dekodowanie i przemieszczanie się po innych poziomach menu konfiguracyjnego wygląda analogicznie od strony programowej.

Najłatwiej zrealizować dynamiczne wyświetlanie, używając licznika, który generuje przerwanie przy przepelnieniu. Wówczas można kontrolować częstotliwość uaktywniania kolejnych linii kolumn. TIMER1 został

Rys. 8 Kody klawiatury PC AT

ESC 7E	F1 05	F2 06	F3 04	F4 0C	F5 03	F6 0B	F7 83	F8 0A	F9 01	F10 09	F11 78	F12 07	E012E07C	E11477E1	F014F077				
~ 0E	1! 16	2@ 26	3# 28	4\$ 25	5% 2E	6^ 38	7& 3D	8* 3E	9(46	0) 45	-+ 55	[\] SD	Pr Scr	Scr L	Pause Bk				
TAB 0D	Q 15	W 1D	E 24	R 2D	T 2C	Y 35	U 3C	J 43	O 44	P 4D	[(54]) 5B	Ins E070	Home E06C	P Up E07D	Num 77	/ E04A	* 7C	- 7B
Caps 5B	A 1C	S 1B	D 23	F 2B	G 34	H 33	J 3B	K 42	L 4B	:: 4C	'' 52	← SA	Del E071	End E069	P Dn E07A	7 6C	8 75	9 7D	+ 79
Shift 12	Z 1A	X 22	C 21	V 2A	B 32	N 31	M 3A	< 41	> 49	7/ 4A	Shift 59	↑ E075	↓ E072	← E068	→ E074	1 69	2 72	3 7A	Enter E05A
Ctrl 14	Alt 11	SPACE 29										Alt E011	Ctrl E014	0 70	1 71				



Rys. 9 Zaświecanie pojedynczej kolumny matrycy

tak skonfigurowany, że przerwanie jest zgłaszane z częstotliwością 4,8kHz. Każda z linii kolumn jest sterowana co 48 cykli wyświetlania, więc częstotliwość, z jaką są zaświecane (gaszone) diody w kolumnach, wynosi 100Hz. Linie mikrokontrolera muszą generować sygnał sterujący zgodnie z przebiegami przedstawionymi na rysunku 9.

Sterowanie układem polega na generowaniu sygnałów dla interfejsu SPI, które realizuje się programowo. Powodem jest przyjęte założenie realizacji urządzenia. W przypadku obsługi poprzez SPI sprzętowe – linię DS należy połączyć z linią MOSI układu master, linię SH_CP z linią SCK, natomiast ST_CP z linią SS.

Zmienna *licznik_zbocze* odpowiada za generowanie sygnałów. Na początku linia sygnału SH_CP jest ustawiana w stan wysoki. W kolejnym wywołaniu obsługi przerwania linie SH_CP oraz ST_CP są w stanie niskim, a na linii DS ustawiany jest stan wysoki. Następne wywołanie przerwania powoduje ustawienie linii SH_CP w stan wysoki. Kolejna obsługa przerwania polega na ustawieniu linii SH_CP w stan niski, linii DS w stan niski, natomiast linii ST_CP w stan wysoki, co powoduje pojawienie się wysokiego poziomu na wyjściu pierwszym (Q0) układu '595 i wysterowanie pierwszej kolumny. Wówczas na PORTD jest podawana odpowiednia liczba, która poprzez tranzystory BC327 zaświeca odpowiednie diody w wierszach dla pierwszej kolumny. Kolejne wywołanie przerwania od TIMER1 powoduje ustawienie linii SH_CP w stan wysoki (przesunięcie wpisu z linii DS o jedną pozycję w prawo), a linii ST_CP w stan niski. Następna obsługa przerwania od TIMERA1 to ustawienie linii SH_CP w stan niski, linii ST_CP w stan wysoki, co powoduje pojawienie się wysokiego poziomu na wyjściu drugim (Q1) układu '595 i wysterowanie kolejnego tranzystora, który zamyka obwód przepływu prądu dla następnej kolumny. Wówczas na PORTD jest podawana odpowiednia liczba, która poprzez tranzystory BC327 zaświeca odpowiednie diody w wierszach dla drugiej kolumny. Analogicznie odbywa się zaświecanie pozostałych kolumn. Układy '595 są połączone (wyjście Q7' z wejściem DS następnego

układu), więc powoduje przesuwanie wpisu z linii DS aż do ostatniego układu. Zmienna *licznik* zlicza liczbę zaświecanych kolumn, która zostaje wyzerowana i cykl zaczyna się od początku. Wyświetlanie realizowane jest od strony lewej do prawej wyświetlacza.

Dana dla PORTD mikrokontrolera sterującego wierszami wyświetlacza jest zwracana przez funkcję *show()*. Jej zadaniem jest pobieranie wartości z tablicy *tekst[]*, które odpowiada wyświetlanemu znakowi. Następnie dla tego znaku są kolejno pobierane liczby z pamięci programu (plik *dane.h*), które odpowiadają za zaświecenie odpowiednich diod w wierszu. Gdy funkcja trafi na liczbę, oznaczającą brak dalszych kolumn dla znaku (0b10101010), wtedy jest pobierany kolejny znak z tablicy *tekst[]* i analogicznie dane odpowiadające zaświeceniu odpowiednich wierszy są zwracane przez funkcję. Aktualnie wyświetlany znak jest przechowywany w zmiennej *litera*, natomiast aktualnie wyświetlana kolumna w zmiennej *kolumna_pamiec*. Identycznie wygląda wyświetlanie dla menu konfiguracyjnych (znaki do wyświetlania są pobierane z odpowiednich tablic, które odpowiadają danemu menu).

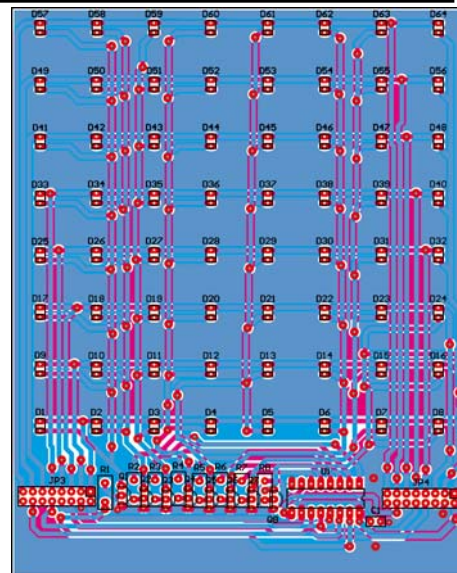
Przesuwanie jest realizowane poprzez odczytywanie znaku od kolumny pierwszej, w następnym przesunięciu od drugiej, w następnym przesunięciu od trzeciej itd. („ucinianie” znaku o jeden pixel więcej niż poprzednio zaczynając od lewej). Gdy jest koniec kolumn w znaku, następuje analogicznie przesuwanie kolejnego znaku. W zmiennej *move_litera* jest przechowywana wartość dziesiątka znaku ASCII, od którego zaczyna się przesuwanie, natomiast zmienna *move_kolumna* przechowyuje numer kolumny, od której zaczyna się wyświetlanie znaku.

Wzory znaków, które są zapisane w pliku *scancodes.h*, zostały wygenerowane za pomocą dodatkowego programu napisanego w języku C# (duży wkład mojego brata). Poprzez naciśnięcie przycisku zaświecamy wirtualną diodę. Po obliczeniu wzoru nowego znaku możemy go zaktualizować ze znajdującym się w tablicy znaków (plik *scancodes.h ()*).

Zmiany i modyfikacje

Urządzenie można rozbudować o liczne funkcje dodatkowe, na przykład:

- Zwiększenie liczby matryc – potencjalny klient zobaczy więcej informacji w tym samym czasie.

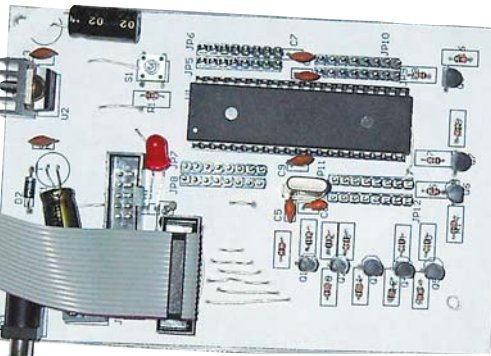
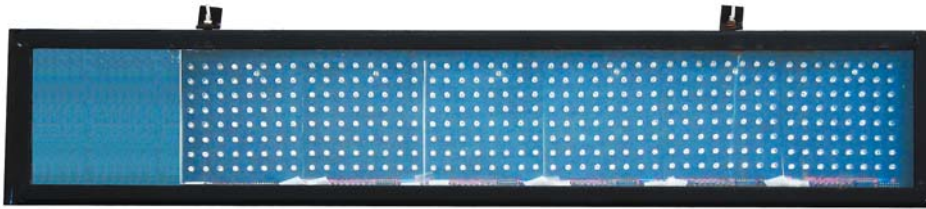


Rys. 10 Płytkę modułu wyświetlacza - skala 50%

- wykorzystanie czcionki o rozmiarze 7 x 5
- zmniejszenie napisów zwiększa długość prezentowanego tekstu.
- Sterowanie układem 74HC595 poprzez szybsze sprzętowe SPI (wtedy trzeba wymienić 74HC595 na szybszy 74AHC595).
- Zwiększenie częstotliwości rezonatora kwarcowego.
- Czas/data/temperatura/orientacja – menu wyświetlacza przewiduje dodanie tych czterech funkcji.
- Sterowanie/wprowadzanie tekstu zdalnie, za pomocą pilota, Bluetooth, RS-232, Wi-Fi, Ethernetu.
- Regulacja jasności świecenia za pomocą PWM.
- Dodanie zewnętrznej pamięci EEPROM.
- dodanie znaczników, sterujących przepływem tekstu (np. przyśpieszanie/zwalnianie/zatrzymanie).

Wykaz elementów	
Moduł matrycy LED	C5-C6 22pF
Rezystory	C2-C3,C7-C9 100nF
R1-R8 4,7kΩ	Półprzewodniki
Kondensatory	Q1-Q8 BC327
C1 100nF	D1 LED
Półprzewodniki	D2 1N4001
Q1-Q8 BC337	D3 1N4007
D1-D64 LED	U1 ATmega16
U1 74HC595	U2 L7805
Pozostałe	Pozostałe
JP1,JP2IDC 16-pin	X1kwarc 8MHz
Sterownik	M1 mostek prostowniczy 1,5A
Rezystory	TRAF0 TS10/35
R1-R8 1kΩ	S1 microswitch
R9-R16 33kΩ	JP1IDC 10-pin
R17 22kΩ	JP2 złącze IDC 16-pin
R18 470Ω	JP5-JP14 goldpiny
Kondensatory	Złącze klawiatury PS/2
C1 470μF	R1-R2 4,7kΩ
C4 1000μF	JP1 goldpiny
	JP2gniazdo PS/2 do druku

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2988.



Montaż i uruchomienie

Płytki drukowane zostały zaprojektowane w programie Protel 99SE (Można je pobrać z Elportalu, jak również pliki PDF). Moduły matrycy 8 x 8 zmontowane są na dwustronnych płytkach drukowanych (rysunek 10). Sterownik jest zmontowany na płycie jednostronnej – rysunek 11.

Płytki modelu zostały wykonane metodą termotransferu (tzw. „żelazkowa”). Przejściówkę – złącze klawiatury PS/2 wykonano na płycie uniwersalnej ze względu na bardzo prostą budowę. Wszystkie elementy są w wersji do montażu przewlekane.

Montaż nie jest trudny, ale wymaga cierpliwości i dokładności. Po wykonaniu płytek drukowalnych należy w pierwszej kolejności wykonać przelotki – wywiercić otwory przelotek. Najprościej przylutować drucik (np. odcięte fragmenty nóżek rezystorów lub diod) z obu stron laminatu. Następnie nanieść soldermaszkę, która zabezpiecza miedziane ścieżki przed utlenianiem oraz poprawia wygląd całego układu. Kolejny krok to dokończenie wiercenia pozostałych otworów, przyłożenie czarnej kartki pod diody (efekt kontrastu z czerwonymi diodami) i włożenie wszystkich 64 diod, które następnie trzeba przylutować. W następnym kroku należy zamontować i przylutować tranzystory, później dokonać montażu pozostałych elementów. Przy montażu układu sterownika warto przyjąć zasadę: od najmniejszego do największego.

Wszystkie moduły matryc 8 x 8, układ złącza klawiatury PS/2 oraz sterownik przykręcono śrubkami do metalowej blachy, którą wcześniej zaizolowano odpo-

wiednią taśmą i połączono krótkimi odcinkami 16-pinowej taśmy. Kolejnym etapem jest zaprogramowanie mikrokontrolera (pamięć flash i eeprom) i podłączenie klawiatury AT oraz zasilania. Na wyświetlaczu powinien pojawić się napis „Witaj”. Ramę wyświetlacza (wymiary układu matrycy oraz całego wyświetlacza zamieszczono w **Instrukcji obsługi urządzenia**) wykonano z metalowego „korytka” o szerokości 4 cm. Pomalowanie farbą koloru czarnego poprawia wygląd oraz zabezpiecza przed

warunkami atmosferycznymi. Następnie przyklejono silikonem przezroczystą pleksi o grubości 3 mm. Wsunęto metalową blachę z modułami i przykręcono ją do ramy za pomocą wkrętów do blachy. Ostatni krok montażu to przykręcenie bocznego fragmentu ramy wyświetlacza. Taka budowa dobrze zabezpiecza układy przed trudnymi warunkami atmosferycznymi.

Montaż wyświetlacza. Przy zastosowaniu diod o wąskim kącie świecenia należy zwrócić szczególną uwagę na prawidłową wysokość zawieszenia i kąt nachylenia wyświetlacza.

Znając odległość (Y), od której wyświetlany tekst powinien być widoczny z maksymalną jasnością świecenia, można obliczyć wysokość zawieszenia wyświetlacza (X). W miejsce Y należy podstawić zmierzoną odległość i pomnożyć przez 0,176. Rysunek techniczny (w formacie PDF) można ściągnąć z Elportalu.

Prezentowany model działa prawidłowo od lipca 2010. Charakteryzuje się zwartą konstrukcją, prostą obsługą, odpornością na warunki atmosferyczne i bardzo dobrą czytelnością, nawet w słoneczne dni. Bardzo dobrze sprawdza się jako atrakcyjna reklama świetlna.

Instrukcja obsługi

Podłączenie i odłączenie klawiatury od wyświetlacza należy wykonać przy włączonym zasilaniu (w przypadku braku reakcji na wciśnięcie klawisza **Esc** należy odłączyć i podłączyć ponownie zasilanie). Urządzenie posiada

pamięć 400 znaków, która pamięta wprowadzony tekst także po odłączeniu zasilania.

1. Programowanie wyświetlacza.

Zmiana treści wyświetlanej informacji:

- 1) Wywołać menu konfiguracyjne, wciskając klawisz **Esc**,
- 2) Klawiszem **1** wybrać opcję „1-Tekst”,
- 3) Dokonać edycji tekstu (szczegóły dalej)
- 4) Zmiany zatwierdzić klawiszem **Enter**.

Zmiana prędkości przepływu tekstu:

- 1) Wywołać menu, wciskając **Esc**,
- 2) Klawiszem **2** wybrać opcję „2-Prędkość”,
- 3) Klawiszami **↑, ↓** wybrać żądaną prędkość przepływu w przedziale 00-99,
- 4) Zmiany zatwierdzić klawiszem **Enter**.

Aby **zwiększyć/zmniejszyć odstęp pomiędzy końcem a początkiem spotu**, należy dodać odpowiednią liczbę spacji.

Powrót do poprzedniego poziomu menu oraz opuszczenie menu jest możliwe za pomocą klawisza **Esc**.

2. Edycja tekstu.

Kursor. Po włączeniu opcji edycji tekstu, kursor (świecąca pionowa belka) umieszczony jest zawsze na końcu wprowadzanego tekstu. Nie ma możliwości przesunięcia kursora.

Wprowadzanie znaków. Wprowadzany znak pojawia się zawsze w miejscu, w którym aktualnie znajduje się kursor. Wprowadzanie znaków odbywa się analogicznie jak w komputerze klasy PC. Dodanie litery „S” polega na przytrzymaniu klawisza **Shift** (lub wciśnięciu klawisza **Caps Lock**) i wciśnięciu klawisza **s**. Wprowadzenie polskiego znaku „ę” dokonuje się poprzez przytrzymanie prawego klawisza **Alt** i wciśnięcie **e**.

Wyświetlacz ma możliwość wprowadzania znaków występujących jedynie w języku polskim. Niemożliwe jest wprowadzanie znaków dodatkowych. Wyświetlacz nie ma także znaków sterujących, które umożliwiałyby zmianę chwilowej prędkości lub zatrzymanie tekstu przez określony czas.

Usuwanie znaków. Istnieje możliwość usunięcia pojedynczego znaku, znajdującego się przed kursorem. Aby dokonać usunięcia znaku, należy wcisnąć klawisz **Backspace**.

Tomasz Gilewski
tomaszgilewski@interia.pl

Literatura:

- [1] Baranowski R., *Mikrokontrolery AVR ATmega w praktyce*, BTC, Warszawa 2005
- [2] Hadam P., *Projektowanie systemów mikroprocesorowych*, BTC, Warszawa 2004
- [3] Jabłoński T., *Graficzny, modułowy wyświetlacz LED*, „Elektronika praktyczna” 3/2010
- [4] Nota katalogowa ATmega16.pdf
- [5] Nota katalogowa 74HC595.pdf
- [6] Nota katalogowa AVR313: Interfacing the PC AT Keyboard.pdf
- [7] Nota katalogowa LL-503VC2E-V1-2BC.pdf
- [8] www.rgbtechnology.pl
- [9] www.elektroda.pl