



Modułowa matryca LED

W artykule opisano projekt wyświetlacza modułowego. Moduły można łatwo łączyć „bokami” dla osiągnięcia pożądanej długości. Ograniczeniem jest obciążalność zasilacza oraz prąd przewodzenia zastosowanych tranzystorów. Liczbę elementów składowych zredukowano do minimum. Przykładowy kod programu, napisany w języku C, pozwala na skrołowanie napisu na matrycy o dowolnej długości, przy czym napis może składać się z dowolnej liczby znaków ograniczonej jedynie wielkością bufora w pamięci.

Rekomendacje: wyświetlacz przyda się do rozmaitych urządzeń informacyjnych, w tym do samodzielnie wykonywanych, niepowtarzalnych reklam świetlnych.

Od pewnego czasu bardzo popularne stały się matryce diodowe, wykorzystywane w wielu dziedzinach. Na ulicach, w witrynach sklepów, na dworcach, przystankach i w innych miejscach coraz częściej można zobaczyć wyświetlacze, w których zastosowano matryce LED. W urządzeniach przeznaczonych do użytku domowego takie wyświetlacze możemy dostrzec w różnego rodzaju zegarach, wskaźnikach temperatury itd. Zastosowań jest wiele.

Przejdźmy zatem do konkretów, do realizacji układowej. Wyświetlacz opisywany w artykule składa się z modułu sterownika oraz modułu wyświetlacza. Na module wyświetlacza zastosowano dwie matryce LED o rozdzielczości 8×8 pikseli. Płytki zaprojektowano w taki sposób, aby ich szerokość była z każdej strony większa o 2 mm od wymiarów matryc. Pozwala to na łatwe i wygodne mocowanie modułów w obudowie bez użycia wkrętów. Po prostu w obudowie frezujemy rowki, w które wsuwamy poszczególne moduły. Szerokość minimalnie większa od zastosowanego laminatu pozwala na pewne i stabilne mocowanie.

Schemat ideowy modułu LED pokazano na **rysunku 1**. Do sterowania diodami

zastosowano układy scalone SCT2110 firmy Star Chips, co bardzo upraszcza schemat gotowej aplikacji. W niektórych rozwiązaniach są używane oddzielne układy scalone jako rejestry przesuwne oraz oddzielne drivery. Układy SCT2110 integrują w obudowie oba układy: zawierają rejestr przesuwny i 8 źródeł prądowych. Dostępne są również układy z szesnastoma wyjściami, ale o mniejszym dopuszczalnym prądzie obciążenia. Rezystory R3 i R4 ustalają maksymalny prąd diod dołączonych do wyjść, przy czym przy zasilaniu 5 V maksymalny prąd na kanał wynosi 160 mA. Obliczając prąd diody w matrycy, należy brać pod uwagę to, że każda z diod w kolumnie świeci przez 1/8 czasu ze względu na multipleksowanie wyświetlacza. Oczywiście nie możemy przekroczyć prądu szczytowego, bo dioda lub wyjście układu ulegną uszkodzeniu. Najlepiej dobrać rezystor doświadczalnie, dołączając w miejscu rezystora na czas uruchomienia modułu potencjometr 10 kV, a po osiągnięciu zadowalających rezultatów mierząc jego rezystancję i zastępując go rezystorem stałym.

Schemat ideowy modułu sterownika wyświetlacza zamieszczono na **rysunku 2**. Wspomniane wcześniej zamontowane

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-----

Podstawowe parametry:

- Złożony z modułu sterownika i wyświetlacza.
- Moduły wyświetlacza mogą być łączone dla osiągnięcia wymaganej rozdzielczości poziomej.
- Rozdzielczość pojedynczego modułu: 8 (pion) × 16 (poziom) pikseli.
- Rodzaj wyświetlacza: LED.
- Napięcie zasilające: sterownik i moduł +5 V DC.
- Prąd zasilający: zależny od liczby użytych modułów, wyświetlanej treści, ustawionej jasności. Maksymalnie 6,9 A wiersz, 160 mA/kolumnę.

Projekty pokrewne na www.media.avt.pl:

AVT-5606	Magik Matrix (EP 10-11/2017)
AVT-3184	Zegar/termometr z wyświetlaczem matrycowym (Edw 11/2017)
AVT-5561	Efektowny sterownik oświetlenia (EP 12/2016)
AVT-2784	Zegar matrycowy (Edw 4/2006)

Uwaga! Elektroniczne zestawy do samodzielnego montażu.

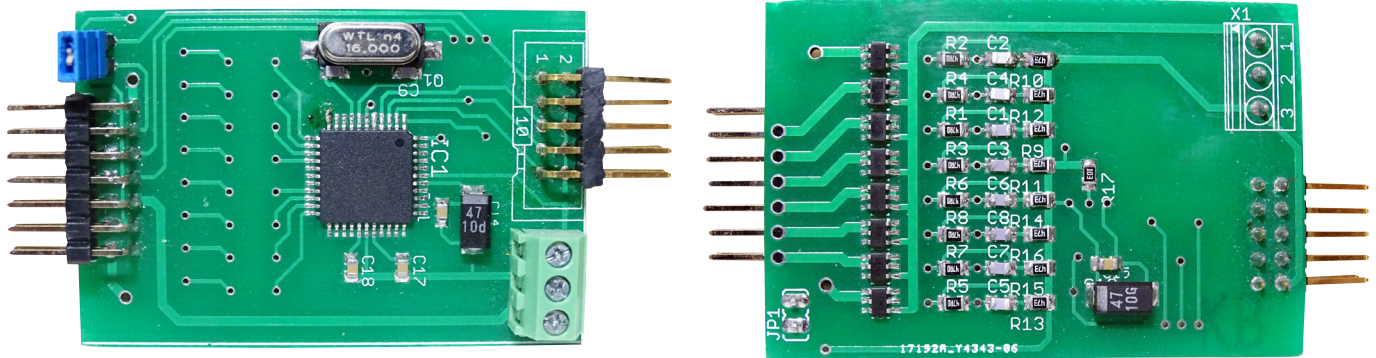
Wymagana umiejętność lutowniaki! Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wzlutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu.

Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wzlutowane w płytkę PCB)
- wersja [A] – płytkę drukowaną bez elementów i dokumentacji Kity w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
 - wersja [A+] – płytkę drukowaną [A] + zaprogramowany układ [UK] i dokumentacja
 - wersja [UK] – zaprogramowany układ

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz: <http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.

na module wyświetlacza układy SCT210 sterują kolumnami. Drivery wierszy zamontowano na płytce sterownika. W ich funkcji zastosowano tranzystory MOSFET-P przystosowane do sterowania napięciem o poziomach TTL, co pozwala na ich bezpośrednie sterowanie z pinu procesora bez użycia dodatkowych driverów. Maksymalny prąd drenu użytych tranzystorów to 6,9 A. Rezystory R1...R8 ograniczają prąd przeładowania

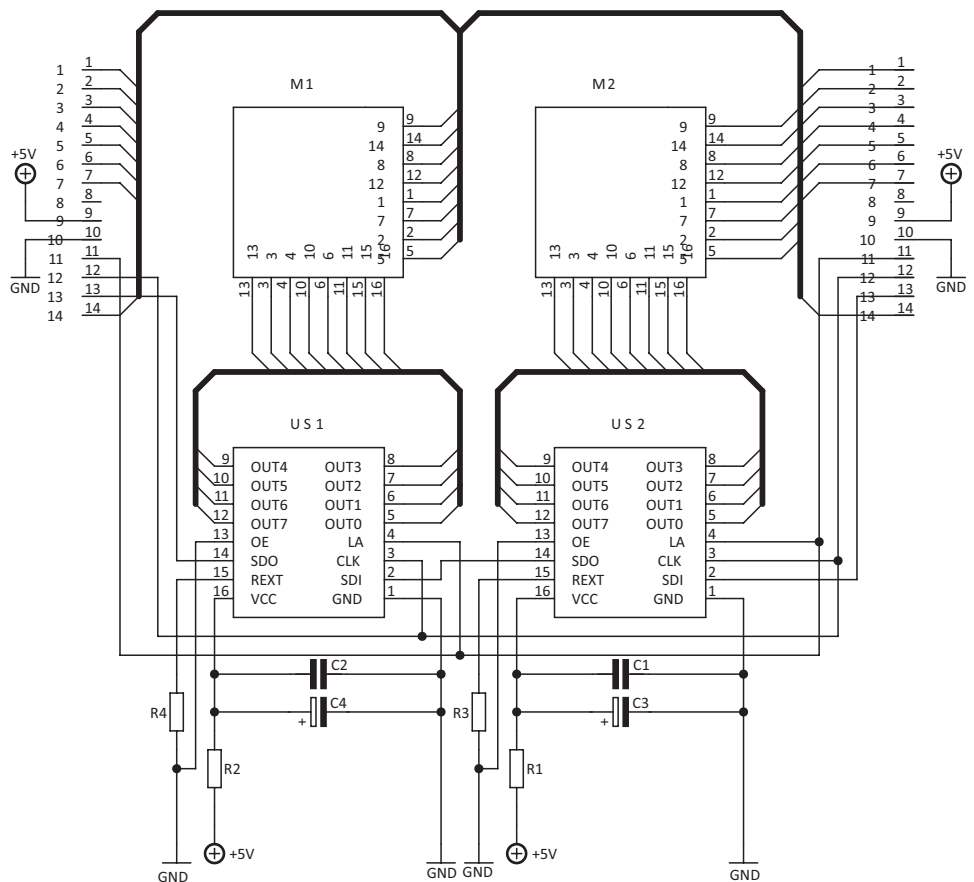


bramki, natomiast kondensatory C1... C8 połączone szeregowo z pojemnością bramki znacząco ten czas skracają. Przy częstotliwości multipleksowania wynoszącej około 400 Hz nie ma to aż tak dużego znaczenia, ale warto zapamiętać ten sposób, ponieważ przy większej częstotliwości przełączania widać różnicę.

Rezystory R9...R16 rozładowują bramki tranzystorów. Zastosowany mikrokontroler nie jest elementem krytycznym – można użyć ATmega32, ATmega644 lub nawet ATmega1284 w obudowie TQFP44. Wszystko zależy od wyświetlanej treści. Dla samych napisów wystarczy ATmega16, ale jeżeli do pamięci Flash załadujemy jakąś bitmapę, to trzeba będzie rozważyć procesor z większą ilością pamięci. Złącza JP1 zapobiega miganiu matryc przy programowaniu. Jak komuś to nie przeszkadza, to można ją zewrzeć na stałe.

Montaż

Schemat montażowy matrycy zamieszczono na rysunku 3, natomiast sterownika na rysunku 4. Montaż jest typowy, ale może przysporzyć problemów ze względu na zastosowane elementy SMD. Dlatego przed jego rozpoczęciem należy wyposażyć się w odpowiednią lutownicę, topnik, plecionkę oraz cienką cynę z dużą ilością topnika. Gęsto rozmieszczone wyprowadzenia można lutować za pomocą grotu „mikrofała”, ale nie jest on niezbędny.

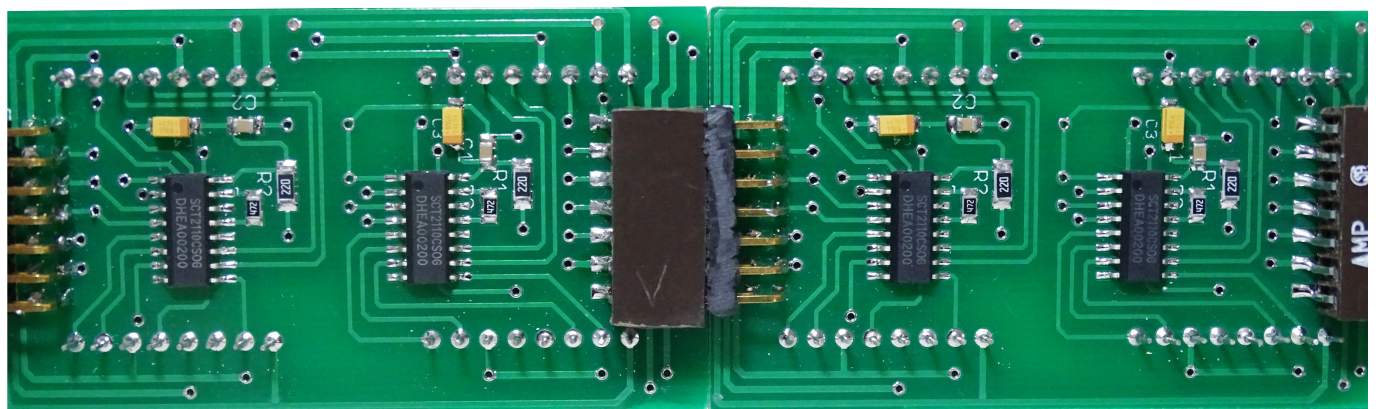


Rysunek 1. Schemat ideowy modułu matrycy wyświetlacza

Oprogramowanie

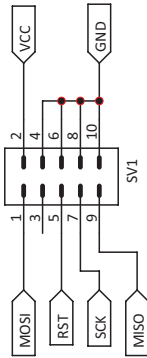
Kluczowe znaczenie dla pracy wyświetlacza ma ustalenie częstotliwości multipleksowania wierszy. Wartość poniżej 30 Hz jest zbyt mała, aby użytkownik nie zauważył migotania. W projekcie przyjęto wartość typową,

taką jak częstotliwość odświeżania ramki w telewizji, wynoszącą 50 Hz. Łatwo policzyć, że przy 8 wierszach daje to 400 Hz. Zasilanie wiersza jest załączane w przerwaniu od 8-bitowego Timera 0. Na listingu 1 pokazano funkcję konfigurującą ten timer.



W pliku nagłówkowym *MATRYCA_1.h* umieszczono wygodne definicje doprowadzeń interfejsu SPI oraz poszczególnych wierszy, a także deklaracje funkcji użytych w programie (**listing 2**).

Jedną z ważniejszych części programu sterującego jest procedura obsługi przerwania. Pokazano ją na **listingu 3**. Na początku powołujemy dwie tablice, za których pomocą będziemy zaświecali właściwy wiersz, a następnie licznik odliczający od 0 do 7. Gasimy wszystkie wiersze dla uniknięcia prześwitów obrazów zwanych niekiedy „duchami”. W następnym kroku wysyłamy dane zaświecając właściwy wiersz, zwiększamy licznik wierszy o 1, a gdy osiągnięta wartość 7, zerujemy go. Podczas sterowania portami zastosowano technikę maskowania nieznaczących



Wykaz elementów:

Płytki modułu matrycy

Rezystory:

R1, R2: 220 Ω
R3*, R4*: 4,7 kΩ

Kondensatory:

C1, C2: 100 nF
C3, C4: 4,7 μF

Półprzewodniki:

US1, US2: SCT2110
M1, M2: LM88HR12-CC lub KWM-30881CVB

Inne:

Złącze męskie, goldpin kątowy 2x7
Złącze żeńskie HARWiN M20-1880746

Płytki modułu sterownika

Rezystory:

R1...R8: 47 Ω
R9...R16: 47 kΩ
R17: 10 kΩ

Kondensatory:

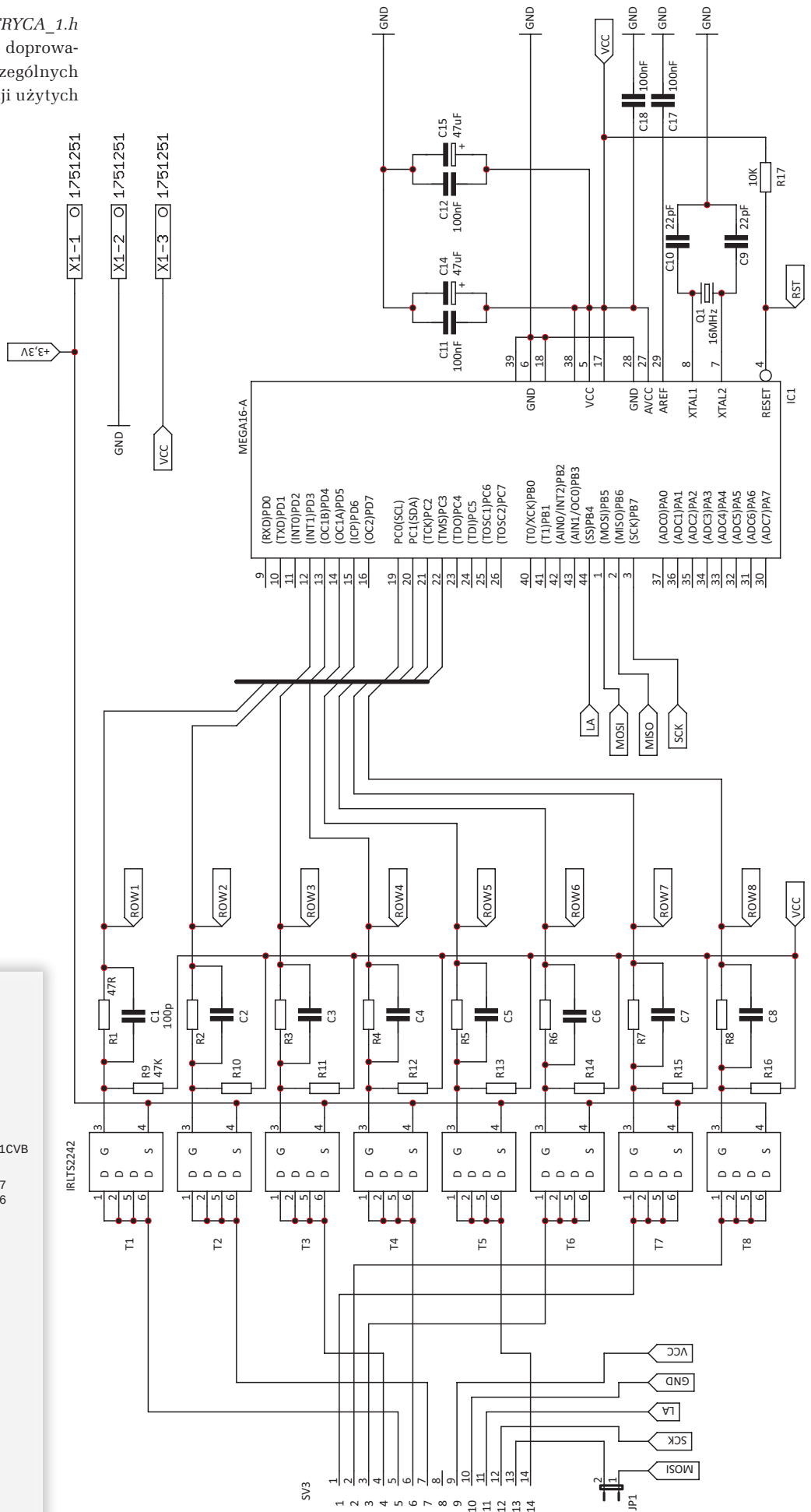
C1...C8: 100 pF
C9, C10: 22 pF
C11, C12, C17, C18: 100 nF
C14, C15: 47 μF

Półprzewodniki:

T1...T8: IRLTS2242
IC1: ATmega16-A

Inne:

Q1 : 16 MHz
SV1: złącze goldpin 2x5 proste
SV3: złącze goldpin 2x7 kątowne
X1: DG381-3.5-03P-14 (Degson Electronics)

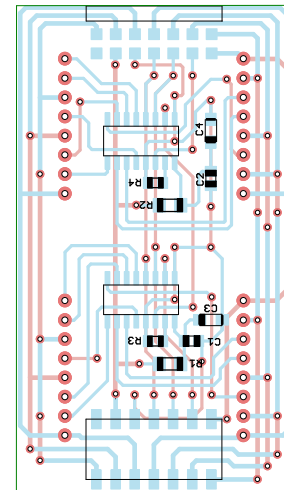


Rysunek 2. Schemat ideowy modułu sterownika wyświetlacza

bitów, to znaczy tych, które nie są modyfikowane przez procedurę obsługi przerwania. Co prawda w tym projekcie nie ma to znaczenia,

ale warto przeanalizować i zapamiętać ten sposób do przyszłych zastosowań.

Funkcję główną programu zamieszczono



Rysunek 3. Schemat montażowy modułu matrycy wyświetlacza

dokonyjemy jedynie operacji na buforze, powodując płynne przesuwanie się napisu po matrycy. Parametr $[k+i]$ powoduje, że napis będzie wsuwany od prawej strony, natomiast jeżeli go zmienimy na $[k-i]$, to napis zostanie wsunięty od lewej strony. Jeżeli chcemy wyświetlić napis statyczny, wystarczy pominąć pierwszą pętlę `for{}`.

Parametr $[k-1]$ określa położenie początku napisu na ekranie. W tym wypadku napis zostanie wyświetlony od drugiej kolumny. Mamy tu duże pole do eksperymentów poprzez modyfikowanie zmiennej i w pierwszej

```
Listing 1. Procedura inicjalizacji Timera 0
void Timer0_Init(void) //włączenie Timera 0
{
    TCCR0 |= (1<<WGM01); //tryb CTC
    TCCR0 |= (1<<CS02); //preskaler 256
    OCR0 = 100; //wartość rejestru dla 400Hz
    TIMSK |= (1<<OCIE0); //zezwoleńie na przerwanie
}
```

```
Listing 2. Plik matryca_1.h
#ifndef MATRYCA_1_H_
#define MATRYCA_1_H_
#define ROWPORT1234 PORTC
#define ROWPORT5678 PORTD
#define ROW1234 DDRC
#define ROW5678 DDRD
#define ROW1234_MASK (ROW1|ROW2|ROW3|ROW4)
#define ROW5678_MASK (ROW5|ROW6|ROW7|ROW8)
#define ROW1 (1<<PC1)
#define ROW2 (1<<PC3)
#define ROW3 (1<<PC0)
#define ROW4 (1<<PC2)
#define ROW5 (1<<PD3)
#define ROW6 (1<<PD6)
#define ROW7 (1<<PD4)
#define ROW8 (1<<PD5)
#define DATAPORT PORTB
#define DATA DDRB
#define DANE (1<<PB5)
#define CLK (1<<PB7)
#define LATCH (1<<PB4)
#define LATCH_ON PORTB |=LATCH
#define LATCH_OFF PORTB &=~LATCH

void Port_Init(void);
void Timer0_Init(void);
void SPI_Init(void);
void Row_Off(void);
void Send_Row(uint8_t RowNumber);
void putChar(uint8_t x, char c, uint8_t *buf);
void putStringP(uint8_t x, const char * s, uint8_t * buf);
void Timer0_Off(void);
uint8_t display[255];
#endif /* MATRYCA_1_H_ */
```

```
Listing 3. Procedura obsługi przerwania od Timer0
ISR(TIMER0_COMP_vect)
{
    const static uint8_t multiplex_tablica1234[] PROGMEM = {~ROW1,~ROW2,~ROW3,~ROW4,255,255,255,255 };
    const static uint8_t multiplex_tablica5678[] PROGMEM = {255,255,255,255,~ROW5,~ROW6,~ROW7,~ROW8 };
    static uint8_t licznik;
    Row_Off();
    Send_Row(licznik);
    ROWPORT1234 =(ROWPORT1234 & ~ROW1234_MASK) | (pgm_read_byte(&multiplex_tablica1234[licznik]) & ROW1234_MASK);
    ROWPORT5678 =(ROWPORT5678 & ~ROW5678_MASK) | (pgm_read_byte(&multiplex_tablica5678[licznik]) & ROW5678_MASK);
    licznik++;
    licznik &= 0x07; // if(licznik>7)licznik=0;
}
```

```
Listing 4. Plik main.c
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <stdlib.h>
#include <avr/pgmspace.h>
#include „MATRYCA_1.h”
#include „fonty.h”

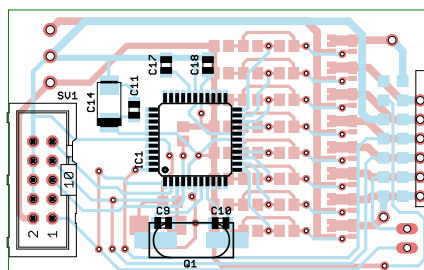
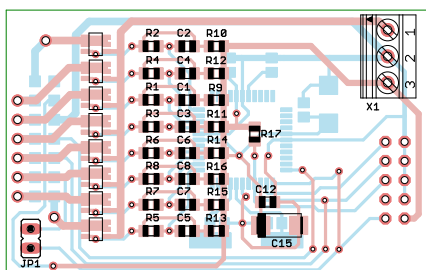
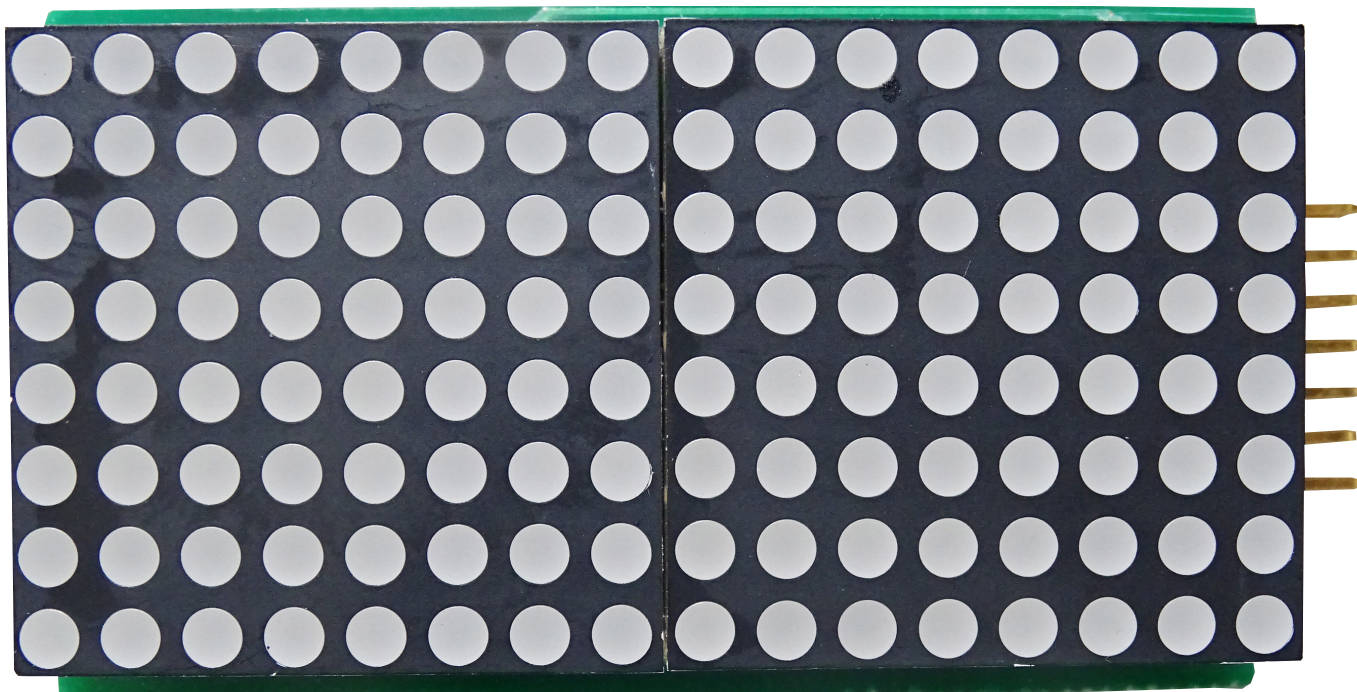
const char PROGMEM Napis1[]={„ELEKTRONIKA PRAKTYCZNA 25 LAT”};
const char PROGMEM Napis2[]={„PRZERWA SERWISOWA - BREAK SERVICE „”};

uint8_t textBufor[64*6];

int main (void)
{
    Port_Init();
    Timer0_Init();
    SPI_Init();
    putStringP(0, Napis1, textBufor);
    sei();
    while(1)
    {
        for (int16_t i=-96; i<156; ++i) // tekst wsuwamy z prawej strony i wysuwamy poza wyświetlacz
        {
            _delay_ms(40); //prędkość przewijania
            for (int8_t k=0; k<96; ++k)
                display[k] = textBufor[k+i];
        }
    }
    while(1)
    {
        for (int8_t k=0; k<96; ++k) display[k] = textBufor[k-1];
    }
}
```

pętli, zmianę opóźnienia prędkości przewijania oraz zmianę pętli głównej, aby – dla przykładu – najpierw wyświetlić napis statyczny, a dopiero po pewnym czasie animować go na ekranie. Można również w trakcie wyświetlania podmienić napis na inny. Eksperymenty pozostawiam zainteresowanym, a w razie problemów służę radą poprzez kontakt e-mail.

W plikach dołączonych do projektu jest przykładowy program źródłowy przesuwały napis na wyświetlaczu „ELEKTRONIKA PRAKTYCZNA 25 LAT”. Plik „fonty.h” zawiera większość używanych znaków, cyfry oraz małe i duże litery. Plik nie zawiera polskich znaków, które trzeba zdefiniować w zależności od potrzeb. Można to zrobić na wiele



Rysunek 4. Schemat montażowy modułu sterownika wyświetlacza

sposobów, ale najłatwiej jest zastąpić nieużywane symbole (na przykład \$, % lub inne) definicjami polskich znaków. Jeśli znak # zastąpimy definicją „L”, to definicja łańcucha „P#YN” spowoduje wyświetlenie napisu „PŁYN”. Wyświetlacz pokazany na fotografii tytułowej składa się z 6 modułów, a więc ma rozdzielczość 96×8 pikseli.

Marek Rębecki
marekrebecki@wp.plcv

REKLAMA

E-prenumerata to:

- najszybszy dostęp do nowego wydania magazynu
- wygodne archiwum na www.avt.pl
- hipertekstowy spis treści i wyszukiwarka
- wbudowane linki – klikasz i jesteś na odpowiedniej stronie WWW

Chcesz otrzymywać dodatek
Niezbędnik Elektronika?
Zamów prenumeratę
drukowaną na
www.avt.pl/prenumerata

Zamów e-prenumeratę (.pdf) na
www.avt.pl/prenumerata/elektroniczne

e-prenumerata roczna z rabatem

15% – 91,80 zł

e-prenumerata dwuletnia z rabatem

30% – 151,20 zł



Prenumeratorki wersji drukowanej za równoległe e-wydania płać tylko 20% ceny: 21,60 zł/rok i 43,20 zł/2 lata