

Rys. 2 Schemat elektryczny wyświetlacza

Rezystor R3 służy do ustalenia maksymalnej wartości prądu płynącego przez diody. Dla podanej wartości prąd ten wynosi około 40 mA. I może być regulowany programowo w zakresie 2,5...40 mA z krokiem 2,5 mA. Zworki JP1 i JP2 służą do ustalenia adresu układu MAX6953 na magistrali I2C.

Montaż

Montaż należy rozpocząć od wlutowania rezystorów, następnie kondensatorów i podstawki pod układ US1 (rozміщення elementów przedstawiono na rys. 3). W następnej kolejności należy wlutować zworki JP1 i JP2 oraz złącze CON1. Po zamontowaniu tych elementów można przejść do montażu wyświetlacza, które są umieszczone od strony ścieżek. Wyświetlacze można wlutować bezpośrednio w płytkę lub umieścić podstawkach wykonanych, na przykład po rozcięciu podstawki precyzyjnej.

Po zamontowaniu wszystkich elementów zworkami JP1 JP2 należy ustawić adres pod jakim będzie się zgłaszał wyświetlacz na magistrali I2C. Poprzez złącze CON1 należy doprowadzić napięcie zasilające wyświetlacz (2,7...5,5 V) oraz sygnały szyny I2C(SCL i SDA).

Sterowanie pracą wyświetlacza

Komunikacja pomiędzy układem MAX6953 a mikrokontrolerem sterującym odbywa się w standardowy sposób, czyli rozpoczyna się od wysłania adresu układu na magistrali I2C, następnie wysyłany jest adres rejestru, do którego ma być zapisana wartość, a na samym końcu należy wysłać tą wartość (rys. 4). Większość poleceń jest wysyłana w takim formacie, a jedynie definiowanie własnych znaków przebiega nieco inaczej.

W tab. 1 przedstawiono adresy, pod którym będzie się zgłaszał wyświetlacz na magistrali I2C w zależności od usta-

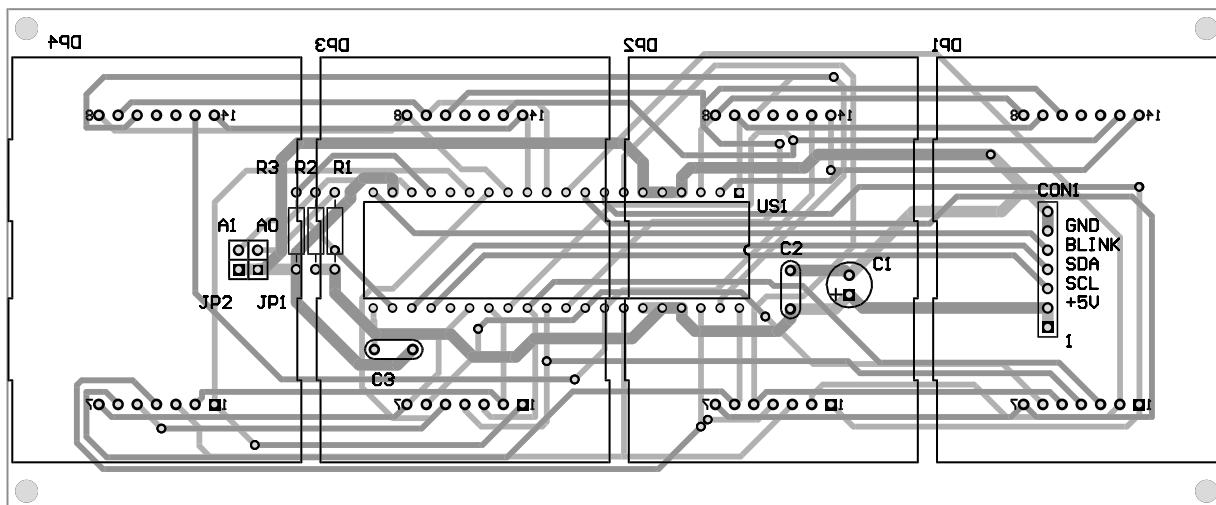
wienia zworek JP1 i JP2 (stan 0 oznacza zwarcie, stan 1 rozwarcie styków).

Jak wspomniano wcześniej w układzie MAX6953 znajduje się generator znaków ASCII, tablica tych

JP2	JP1	Adres
0	0	10100000
0	1	10100010
1	0	10101000
1	1	10101010

znaków jest przedstawiona na rys.5. Oprócz znaków stałych można zdefiniować 24 własne znaki, które umieszczone zostaną w pamięci RAM układu. Wyświetlenie zdefiniowanego znaku odbywa się poprzez wysłanie do wyświetlacza wartości z przedziału 0...23 (pozycje RAM0...RAM23 w tablicy znaków).

Do konfiguracji wszystkich parametrów wyświetlanych znaków służy 20 rejestrów zawartych w układzie



Rys. 3. Rozmieszczenie elementów na płytce wyświetlacza

Tab. 2. Adresy i funkcje rejestrów układu MAX6953

Rejestr	Adres komendy								Wartość HEX
	D15	D14	D13	D12	D11	D10	D9	D8	—
Nie używany	X	0	0	0	0	0	0	0	0x00
Intensywność znaku 10	X	0	0	0	0	0	0	1	0x01
Intensywność znaku 32	X	0	0	0	0	0	1	0	0x02
Ilość znaków	X	0	0	0	0	0	1	1	0x03
Konfiguracja	X	0	0	0	0	1	0	0	0x04
Definicja znaków użytkownika	X	0	0	0	0	1	0	1	0x05
Zarezerwowany	X	0	0	0	0	1	1	0	0x06
Test wyświetlacza	X	0	0	0	0	1	1	1	0x07
Znak 0 bank P0	X	0	1	0	0	0	0	0	0x20
Znak 1 bank P0	X	0	1	0	0	0	0	1	0x21
Znak 2 bank P0	X	0	1	0	0	0	1	0	0x22
Znak 3 bank P0	X	0	1	0	0	0	1	1	0x23
Znak 0 bank P1	X	1	0	0	0	0	0	0	0x40
Znak 1 bank P1	X	1	0	0	0	0	0	1	0x41
Znak 2 bank P1	X	1	0	0	0	0	1	0	0x42
Znak 3 bank P1	X	1	0	0	0	0	1	1	0x43
Jednoczesny zapis do banków P0 i P1 cyfry 0 (odczyt zawsze zwraca 0)	X	1	1	0	0	0	0	0	0x60
Jednoczesny zapis do banków P0 i P1 cyfry 1 (odczyt zawsze zwraca 0)	X	1	1	0	0	0	0	1	0x61
Jednoczesny zapis do banków P0 i P1 cyfry 2 (odczyt zawsze zwraca 0)	X	1	1	0	0	0	1	0	0x62
Jednoczesny zapis do banków P0 i P1 cyfry 3 (odczyt zawsze zwraca 0)	X	1	1	0	0	0	1	1	0x63

MAX6953. Ich adresy oraz funkcje są przedstawione w **tab. 2**.

Funkcje rejestrów są następujące: *Intensywność znaku 10*, adres 01h - zawiera informacje o intensywności świecenia wyświetlaczy 1 i 0. Intensywność może być zmieniana w szesnastu krokach, dlatego dla każdego wyświetlacza jest przeznaczona jedna tetrada bajtu- starsza dla wyświetlacza numer 1, młodsza dla wyświetlacza 0. Wpisanie wartość 0Fh spowoduje, że wyświetlacz 1 będzie świecił z minimalną intensywnością, natomiast wyświetlacz 0 z maksymalną.

Intensywność znaku 32, adres 02h - działanie jak poprzednio, ale dotyczy wyświetlacza 3 i 2.

Ilość znaków, adres 03h - określa czy układ będzie sterował czterema wyświetlaczami jednokolorowymi czy dwoma dwukolorowymi.

Konfiguracja, adres 04hex - rejestr umożliwi ustalenie trybu pracy układu MAX6953.

Definicja znaków użytkownika, adres 05h - po wybraniu tego rejestru można przejść do zapisu w pamięci RAM definiowanych znaków.

Test wyświetlacza, adres 07h - rejestr umożliwia test wyświetlacza poprzez zapalenie wszystkich punktów.

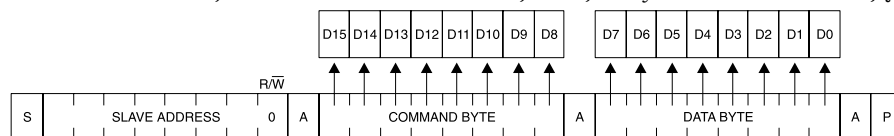
Znak 0 bank P0, adres 20h - rejestr należy do grupy rejestrów, których znajduje się informacja jaki znak ma być wyświetlony na danym wyświetlaczu. Rejestry są podzielone na dwa banki P0 i P1. Do każdego banków można zapisywać wartości oddzielnie lub jednocześnie do obu tą samą wartość. Zapis do rejestrów o adresach 20...23h powoduje wpis znaków do banku P0, natomiast do rejestrów o adresach 40...43h powoduje wpis do banku P1. W układzie MAX6953 istnieją jeszcze dodatkowe cztery rejestry o adresach 60...63h. Wpis do tych rejestrów jest jednoznaczny z wpisaniem tej samej wartości do rejestrów banku P0 i P1.

Podzielenie pamięci znaków na dwa banki nieco komplikuje obsługę wyświetlacza, ale daje także dodatkowe

możliwości możliwe do wykorzystania po wprowadzeniu wyświetlacza w tryb migania (*blink*). Wynika to z faktu, że funkcja *blink* nie powoduje tylko zapalania i wygaszania wyświetlacza lecz naprzemiennie kieruje na wyświetlacz dane z banku P0 i P1. Co to oznacza?

Duża liczba funkcji ułatwia obsługę wyświetlacza, jednak utrudnia „oswojenie” się z układem. Aby ułatwić Czytelnikom zastosowanie prezentowanego wyświetlacza we własnych konstrukcjach na **list. 1** przedstawiono gotowe procedury do jego obsługi. Procedury napisane są w języku C, jednak na ich podstawie można stworzyć własne przystosowane do innego języka programowania. W procedurach tych obsługa magistrali I2C została pominięta i fizyczna obsługa portów mikrokontrolera muszą wykonywać dodatkowe procedury.

Procedura `MAX6953_init(void)` inicjalizuje wyświetlacz ustawiając



Rys. 4. Format danych wysyłanych do układu MAX6953

Tab. 3 Przykład definiowania własnych znaków

Znak w tablicy znaków	Numer definiowanego znaku	Adres w pamięci znaków (HEX)	Dane w pamięci znaków							
			D7	D6	D5	D4	D3	D2	D1	D0
RAM00	0	0x80	0	0	1	1	1	1	1	0
RAM00		0x81	0	1	0	1	0	0	0	1
RAM00		0x82	0	1	0	0	1	0	0	1
RAM00		0x83	0	1	0	0	0	1	0	1
RAM00		0x84	0	0	1	1	1	1	1	0
RAM01	1	0x85	0	0	0	0	0	0	0	0
RAM01		0x86	0	1	0	0	0	0	1	0
RAM01		0x87	0	1	1	1	1	1	1	1
RAM01		0x88	0	1	0	0	0	0	0	0
RAM01		0x89	0	0	0	0	0	0	0	0
RAM02	2	0x8A	0	1	0	0	0	0	1	0
RAM02		0x8B	0	1	1	0	0	0	0	1
RAM02		0x8C	0	1	0	1	0	0	0	1
RAM02		0x8D	0	1	0	0	1	0	0	1
RAM02		0x8E	0	1	0	0	0	1	1	0

go w tryb aktywny, wyświetlacze zostają wygaszone, zostaje ustawiona maksymalna intensywność świecenia. Procedurę tą należy

wywołać na samym początku obsługi wyświetlacza.

Procedura `LED_putc(char character, char digit_position)` umożliwia

wyświetlanie na wyświetlaczu znaku na podanej pozycji. Jako `digit_position` można podać pozycję zapisu do banku P0, P1 lub do obu jednocześnie.

Procedura `MAX6953_LED_clear()` czyści cały wyświetlacz wpisując na wszystkie pozycje spacje.

`MAX6953_test_on()` włącza tryb testu wyświetlacza, a `MAX6953_test_off()` wyłącza ten tryb.

Procedura `MAX6953_blink_on()` włącza tryb mrugania wyświetlacza lub przemiennego wyświetlania znaków zapisanych w bankach P0 i P1. „`MAX6953_blink_off()`” wyłącza funkcję mrugania wyświetlacza.

Ostatnią funkcją jest funkcja „`MAX6953_char_def(char char_nr, char def0, char def1, char def2, char def3, char def4)`” i umożliwia definiowanie własnych znaków. Wywołując tą funkcję należy podać aż sześć parametrów: pierwszy określa, który znak będzie definiowany (od 0...23), a kolejne pięć bajtów określa, które punkty wyświetlacza będą zapalone. Na podstawie numeru definiowanego znaku procedura oblicza adres w pamięci RAM układu MAX6953 (obszar definiowanych znaków rozpoczyna się od adresu 80hex) i zapisuje do kolejnych komórek parametry definiowanego znaku. Przy definiowaniu własnych znaków najstarszy bit każdego bajtu należy ignorować, gdyż na wyświetlaczu jest wyświetlane tylko siedem młodszych bitów, przy czym najmłodszy bit oznacza punkt oznaczający lewy górny róg wyświetlacza. Przykład definiowania znaków przedstawiono w **tab.3**.

Krzysztof Pławiuk, EP

WYKAZ ELEMENTÓW

Rezystory:

R1, R2: 10 kΩ

R3: 56 kΩ

Kondensatory

C1: 220 μF/16 V

C2: 100 nF

C3: 27 pF

Półprzewodniki

US1: MAX6953

Inne

CON1: Goldpin 1x7 męski

JP1, JP2: Goldpin 1x2 męski + zworka

DP1...DP4: TA20-11EWA

Podstawka DIP40

```
List. 1. Procedury obsługi wyświetlacza
#define LED_ADDR 0b10100000
//adres układu na szynie I2C

#define DIGIT_0_1_INT_REG 0x01
//adres rejestru 0
#define DIGIT_2_3_INT_REG 0x02
//adres rejestru 0
#define CONFIG_REG 0x04
#define char_def_reg 0x05
#define TEST_REG 0x07
//test wyświetlacza
#define DIGIT_0_P0 0x20
//adres rejestru cyfry 0
#define DIGIT_1_P0 0x21
//adres rejestru cyfry 1
#define DIGIT_2_P0 0x22
//adres rejestru cyfry 2
#define DIGIT_3_P0 0x23
//adres rejestru cyfry 3

#define DIGIT_0_P1 0x40
//adres rejestru cyfry 0
#define DIGIT_1_P1 0x41
//adres rejestru cyfry 1
#define DIGIT_2_P1 0x42
//adres rejestru cyfry 2
#define DIGIT_3_P1 0x43
//adres rejestru cyfry 3

#define DIGIT_0 0x60
//adres rejestru cyfry 0
#define DIGIT_1 0x61
//adres rejestru cyfry 1
#define DIGIT_2 0x62
//adres rejestru cyfry 2
#define DIGIT_3 0x63
//adres rejestru cyfry 3
#define CHAR_DEF_START 0x80 //początek pamięci znaków

Led_putc(char character, char digit_position);
void MAX6953_init(void);
void MAX6953_write(char slave_addr, char command_byte, char data_byte);
void MAX6953_LED_clear();
void MAX6953_test_on(void);
void MAX6953_test_off(void);
void MAX6953_char_def(char char_nr, char def0, char def1, char def2, char def3, char def4);
void MAX6953_blink_on();
void MAX6953_blink_off();
//*****
/* Inicjalizacja wyświetlacza */
//*****
void MAX6953_init(void)
{
    MAX6953_write(LED_ADDR, CONFIG_REG, 0x01);
    //tryb aktywny
    MAX6953_write(LED_ADDR, DIGIT_0_1_INT_REG, 0xFF); //maksymalna moc dla wszystkich
    MAX6953_write(LED_ADDR, DIGIT_2_3_INT_REG, 0xFF); //wyświetlaczy
    MAX6953_test_off();
    MAX6953_LED_clear();
    //czyszczy wyświetlacz
}
//*****
/* Zapis komendy i danej */
//*****
char MAX6953_write(char slave_addr, char command_byte, char data_byte)
{
    I2C_start(); //I2C start
    i2c_write(slave_addr & 0xfe); //adres układu
    i2c_write(command_byte); //komenda
    i2c_write(data_byte); //dana
    i2c_stop(); //I2C stop
}
//*****
/* Zapis znaku na podany wyświetlacz */
//*****
void LED_putc(char character, char digit)
{
    MAX6953_write(LED_ADDR, digit, character);
}
```

```
List. 1. c.d.
}
//*****
/* Włącza tryb testowy- zapala wszystkie wyświetlacze */
//*****
void MAX6953_test_on(void)
{
    MAX6953_write(LED_ADDR, TEST_REG, 0x01);
}
//*****
/* Wylacza tryb testowy- zapala wszystkie wyświetlacze */
//*****
void MAX6953_test_off(void)
{
    MAX6953_write(LED_ADDR, TEST_REG, 0x00);
}
//*****
/* Włącza mruganie wyświetlacza */
//*****
void MAX6953_blink_on(void)
{
    MAX6953_write(LED_ADDR, CONFIG_REG, 0x09);
}
//*****
/* Włącza mruganie wyświetlacza */
//*****
void MAX6953_blink_off(void)
{
    MAX6953_write(LED_ADDR, CONFIG_REG, 0x01);
}
//*****
/* Definiowanie własnych znakow, nalezy podac numer znaku 0-23, nastepnie 5 bajtow, znaczące jest 7 najmłodszych bitow */
//*****
void MAX6953_char_def(char char_nr, char def0, char def1, char def2, char def3, char def4)
{
    //oblicza adres w RAM na podstawie numeru znaku
    MAX6953_write(LED_ADDR, char_def_reg, ((char_nr*5)+char_def_START));
    //zapisuje definiowany znak zerując najstarszy bit
    MAX6953_write(LED_ADDR, char_def_reg, def0 & 0x7F);
    MAX6953_write(LED_ADDR, char_def_reg, def1 & 0x7F);
    MAX6953_write(LED_ADDR, char_def_reg, def2 & 0x7F);
    MAX6953_write(LED_ADDR, char_def_reg, def3 & 0x7F);
    MAX6953_write(LED_ADDR, char_def_reg, def4 & 0x7F);
}
//*****
/* Zapisuje spacje na wszystkie wyświetlacze */
//*****
void MAX6953_LED_clear()
{
    MAX6953_write(LED_ADDR, DIGIT_0, , ,);
    MAX6953_write(LED_ADDR, DIGIT_1, , ,);
    MAX6953_write(LED_ADDR, DIGIT_2, , ,);
    MAX6953_write(LED_ADDR, DIGIT_3, , ,);
}
//*****
```