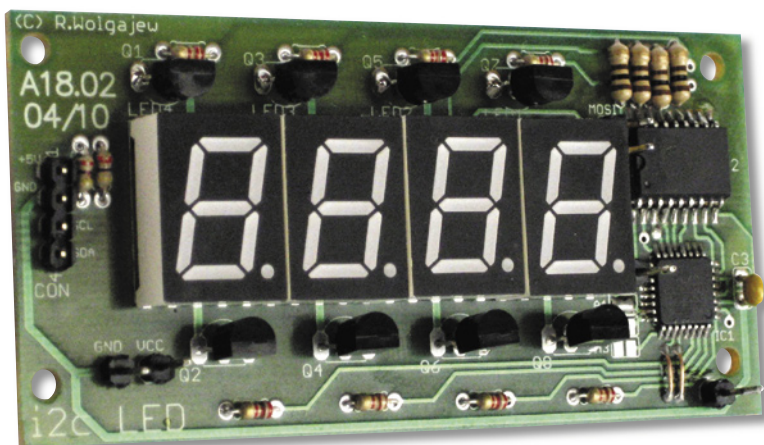


# Moduł wyświetlacza LED sterowany magistralą I<sup>2</sup>C

W tak zwanym interfejsie użytkownika współczesnych urządzeń elektronicznych stosowane są powszechnie różnego rodzaju wyświetlacze: alfanumeryczne i graficzne LCD, wskaźniki 7-segmentowe LED, OLED itp., a nawet pojedyncze diody świecące. Dla wygody wskaźniki 7-segmentowe często są łączone w moduły. W EP prezentowaliśmy już wiele wersji takiego rozwiązania, ale wydaje się, że pomysłowość konstruktorów nie ma granic.

**Rekomendacje:** projekt przydatny w aplikacjach, w których zachodzi konieczność wyświetlania informacji numerycznej (w tym heksadecymalnej) na wielu niezależnych modułach sterowanych przez jeden system mikroprocesorowy.



W wielu systemach mikroprocesorowych zachodzi potrzeba użycia bardzo czytelnych i zarazem niedrogich 7-segmentowych wyświetlaczy LED i jednoczesnej prezentacji dużej ilości danych. O ile sterowanie tego typu wyświetlaczami jest bardzo proste, o tyle użycie większej liczby tego typu elementów stwarza wiele problemów i niedogodności. Nawet przy sterowaniu multipleksowym, użycie dużej liczby wyświetlaczy LED zajmie cenne porty mikrokontrolera oraz obciąży go programowo, może także znacznie ograniczyć jasność świecenia diod LED, z których zbudowany jest każdy segment modułu. Idealnym, jak się wydaje, rozwiązaniem jest zastosowanie odrębnych modułów wyświetlaczy wyposażonych we własny sterownik i sterowanych ze wspólnej magi-

strali systemowej sterownika głównego. Takim przecież sterownikiem jest dla przykładu każdy moduł wyświetlacza LCD. Rozwiązanie to znajduje także uzasadnienie ekonomiczne, gdyż przy obecnych cenach popularnych mikrokontrolerów, jest ono niedrogą alternatywą dla skomplikowanych systemów jednoukładowych. Proszę wszakże zwrócić uwagę na schemat niniejszego urządzenia (rys. 1) i zauważyć, iż poza tranzystorami sterującymi wspólnymi anodami wyświetlaczy LED, jak i ich połączonymi katodami (układ ULN2803), układ sterownika składa się wyłącznie z jednego elementu – mikrokontrolera ATmega8 pełniącego rolę odbiornika rozkazów magistrali I<sup>2</sup>C i jednocześnie jednostki sterującej pracą wyświetlaczy.

Przedstawione urządzenie posiada następujące cechy:

- możliwość wyświetlania liczb w zapisie dziesiętnym (0...9999),
- możliwość wyświetlania liczb w zapisie heksadecymalnym (0...FFFF),
- możliwość wyświetlania wartości heksadecymalnych (0...F) oddzielnie dla każdej cyfry,
- niezależne sterowanie kropkami dziesiętymi,
- możliwość programowego sterowania kolorem świecenia każdej cyfry modułu (czerwony, zielony, pomarańczowy),
- możliwość programowego sterowania jasnością świecenia całego modułu LED,

## AVT-5147

W ofercie AVT jest dostępna:  
– [AVT-5147A] – płytka drukowana

### PODSTAWOWE PARAMETRY

- Płytko o wymiarach 82x40x20 mm
- Zasilanie 5 V
- Maksymalny prąd obciążenia: 200 mA
- Maksymalna częstotliwość pracy magistrali I<sup>2</sup>C: 400 kHz
- Liczba możliwych adresów sprzętowych: 8 + obsługa wywołania ogólnego
- Liczba rozkazów: 12
- Pamięć ustawień i wartości
- 3 kolory wyświetlania
- Regulacja jasności świecenia

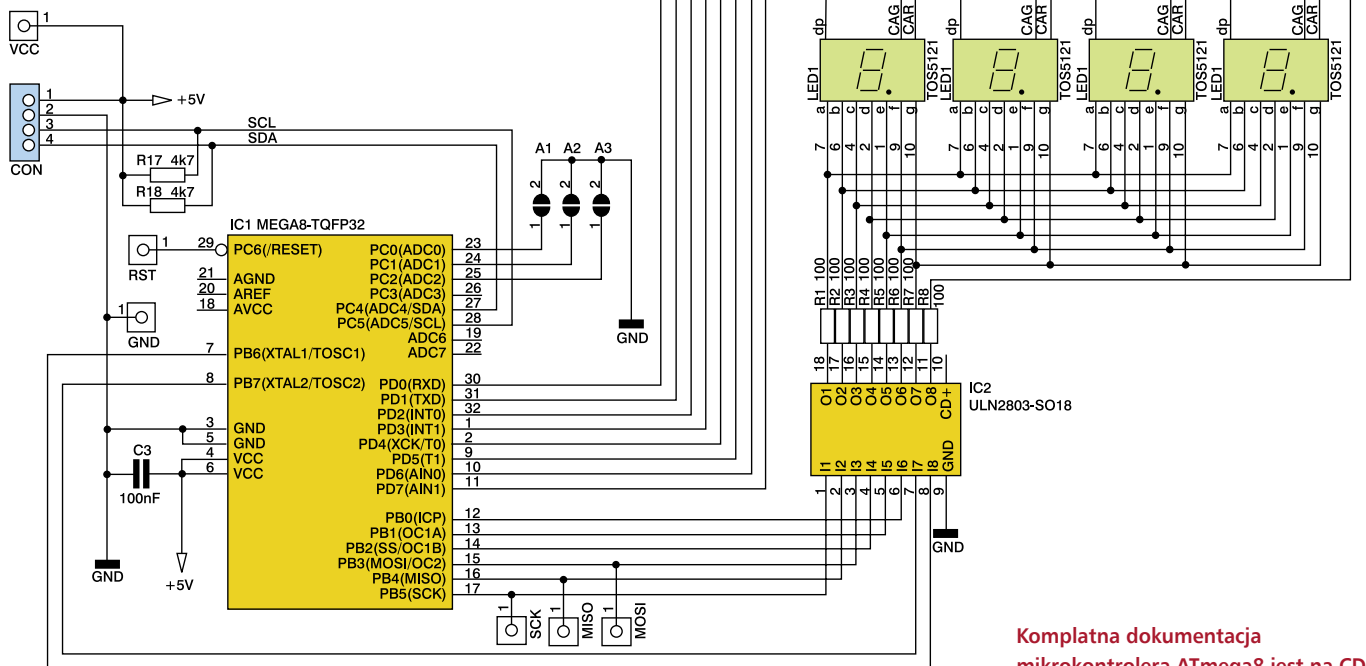


### PROJEKTY POKREWNE

wymienione artykuły są w całości dostępne na CD

Tytuł artykułu	Nr EP/EdW	Kit
Wyświetlacz siedmio-segmentowy JUMBO	EdW 4/1997	AVT-2222
Wielkogabarytowy wyświetlacz siedmio-segmentowy	EP 7/1998	AVT-1186
Podwójny wyświetlacz siedmio-segmentowy sterowany I <sup>2</sup> C	EP 8/2000	AVT-859
8-cyfrowy wyświetlacz LED z interfejsem SPI	EP 6/2006	AVT-934

- obsługa i wyświetlanie informacji o błędach (ramki transmisji, magistrali danych, wartości danych),
- możliwość zapisywania wartości i ustawień (w tym stanów kropek dziesiętnych, jasności świecenia i kolorów poszczególnych cyfr) w pamięci nieulotnej,
- możliwość sprzętowej zmiany adresu (do 8 modułów na jednej magistrali I<sup>2</sup>C),



Rys. 1. Schemat elektryczny układu

Kompletna dokumentacja mikrokontrolera ATmega8 jest na CD



- obsługa tzw. wywołania ogólnego (*general call*) umożliwiającego sterowanie pracą wielu modułów o różnych adresach w tym samym czasie, przy użyciu adresu 0x00,
- zintegrowany interfejs programatora SPI.

W opisywanym układzie zastosowano dwukolorowe, 7-segmentowe wyświetlacze LED z niezależnymi wyprowadzeniami wspólnej anody dla każdej z barw (wyprowadzenia 8 i 3 elementu). Dodatkowym atutem zastosowanych elementów jest możliwość jednoczesnej pracy obu diod świecących znajdujących się w każdym segmencie wyświetlacza, co w rezultacie daje nam możliwość uzyskania trzeciego koloru świecenia, w tym przypadku pomarańczowego. Jak widać, zastosowano multipleksowe sterowanie każdym z modułów LED. Mikrokontroler, w odpowiedzi na dane otrzymane przy pomocy magistrali I<sup>2</sup>C, załącza wspólne anody kolejnych modułów LED „wystawiając” wcześniej odpowiednie wartości do portu PB mikrokontrolera, który to steruje

połączonymi katodami wyświetlaczy. Jeśli przełączanie będzie odpowiednio szybkie (w tym przypadku odbywa się z częstotliwością ok. 100 Hz na każdy moduł LED), zapobiegamy zjawisku migotania wyświetlanych informacji również w przypadku filmowania ich przy użyciu kamery wideo. Dodatkowo, na płytce sterownika przewidziano specjalne punkty lutownicze przeznaczone do sprzętowego ustawienia adresu wyświetlacza w przestrzeni adresowej magistrali I<sup>2</sup>C. Adres ten wyznaczamy według poniższego zapisu (zapis binarny):

MSB							LSB
0	1	0	0	A3	A2	A1	0

gdzie A3, A2 i A1 to stan wyprowadzeń portów PC2, PC1 i PC0 mikrokontrolera ATmega8 (logiczne zero dla zwartych odpowiednich punktów lutowniczych).

Należy dla porządku dodać, iż w celu realizacji funkcji odbiornika magistrali I<sup>2</sup>C wykorzystano wbudo-

wany w mikrokontroler sprzętowy interfejs TWI, który jest niczym innym jak zaawansowanym, sprzętowym interfejsem I<sup>2</sup>C w wykonaniu firmy Atmel. Wszelkie informacje przesyłane magistralą są odczytywane przez program obsługi przerwania od układu TWI, a następnie w pętli głównej programu wyświetlane na modułach LED. Każdy bajt przesyłany przez nadajnik (układ Master) jest potwierdzany sygnałem ACK przez nasz sterownik pełniący rolę układu Slave. Do-



Tab. 1. Lista rozkazów modułu wyświetlacza			
Nr	Opis	Liczba danych	Znaczenie danych
0x00	Wyświetl liczbę w zapisie dziesiętnym	Dana(1)... Dana(3)* * - Dana(3) jest bajtem opcjonalnym, który może zostać pominięty	Wyświetla liczbę w zapisie dziesiętnym (bez zer nieznaczących, wartość 0...9999), gdzie Dana(1) to starszy bajt tej liczby, a Dana(2) to młodszy bajt tej liczby. Dana(3) jest parametrem opcjonalnym i jeśli jest podana, określa kolor wyświetlanej liczby według wzorca (aktywny stan „1”): B7 – zielony składnik cyfry nr 4 B6 – zielony składnik cyfry nr 3 B5 – zielony składnik cyfry nr 2 B4 – zielony składnik cyfry nr 1 B3 – czerwony składnik cyfry nr 4 B2 – czerwony składnik cyfry nr 3 B1 – czerwony składnik cyfry nr 2 B0 – czerwony składnik cyfry nr 1 W przypadku pominięcia tego bajtu, zostanie wyświetlona liczba w kolorze zielonym. Kolor pomarańczowy uzyskujemy poprzez jednoczesne użycie składnika zielonego i czerwonego.
0x01	Wyświetl liczbę w zapisie dziesiętnym i zapamiętaj	j.w.	Jak funkcja 0x00 z jednoczesnym zapamiętaniem wszystkich wartości w nieulotnej pamięci EEPROM. Zapamiętana wartość zostanie wyświetlona po włączeniu zasilania sterownika (łącznie z uwzględnieniem informacji o kolorze świecenia poszczególnych cyfr, jasności wyświetlacza i stanem pracy kropek dziesiętnych).
0x02	Wyświetl liczbę w zapisie heksadecymalnym	j.w.	Jak funkcja 0x00 z tą różnicą, że wyświetla liczbę w zapisie heksadecymalnym z zakresu 0...FFFF
0x03	Wyświetl liczbę w zapisie heksadecymalnym i zapamiętaj	j.w.	Jak funkcja 0x01 z tą różnicą, że wyświetla liczbę w zapisie heksadecymalnym z zakresu 0...FFFF
0x04	Wyświetl cyfrę w zapisie heksadecymalnym	Dana(1)	Wyświetla cyfrę w zapisie heksadecymalnym (0...F) na wybranej pozycji wyświetlacza. Poszczególne bity mają następujące znaczenie: B7...B6 – numer cyfry (licząc od prawej tj. 3 cyfra 4, 2 cyfra 3 itd.) B5 – kolor cyfry (składnik zielony) B4 – kolor cyfry (składnik czerwony) B3...B0 – cyfra Wyświetlana cyfra nie wpływa na zawartość pozostałych segmentów LED.
0x05	Wyświetl cyfrę w zapisie heksadecymalnym i zapamiętaj	Dana(1)	Jak funkcja 0x04 z jednoczesnym zapamiętaniem tej cyfry w nieulotnej pamięci EEPROM (zapamiętany zostaje także stan kropek dziesiętnych dla wszystkich cyfr oraz kolor świecenia). Funkcja nie wpływa na zawartość nieulotnej pamięci dla pozostałych segmentów wyświetlacza.
0x06	Wyświetl kropkę dziesiętną	Dana(1)	Wyświetla kropkę dziesiętną. Poszczególne bity mają następujące znaczenie: B7...B6 – numer kropki (licząc od prawej, jak poprzednio) B5 – kolor kropki (składnik zielony) B4 – kolor kropki (składnik czerwony) B3...B0 – bez znaczenia W przypadku, gdy wyświetlana aktualnie cyfra ma inny kolor, zostanie on zmieniony według specyfikacji koloru kropki dziesiętnej. Wyświetlenie kropki jest możliwe wyłącznie wtedy, gdy wybrany moduł LED wyświetla już jakąś informację.
0x07	Wyłącz kropkę dziesiętną	Dana(1)	B0...B1 określają numer wyłączanej kropki. Funkcja nie wpływa na zawartość wyświetlacza dla pozostałych segmentów oraz na wyświetlaną cyfrę dla segmentu, dla którego zostaje wyłączana kropka.
0x64	Wyłącz wyświetlacz	Nie dotyczy	Wyłącza cały wyświetlacz LED z jednoczesnym wykasowaniem zawartości bufora zmiennych (dla poszczególnych cyfr i kropek) i wyczyszczeniem pamięci EEPROM wszystkich zmiennych (nie dotyczy informacji o jasności świecenia wyświetlacza).
0x65	Wyłącz cyfrę	Dana(1)	Wyłącza wybrany moduł LED (0...3 odpowiada modułom 1...4). Zawartość pozostałych modułów LED pozostaje niezmienniona.
0x66	Ustaw jasność wyświetlacza	Dana(1)	Ustawia jasność wyświetlacza LED w zakresie 0...31 (max). Wartość zostaje zapamiętana w nieulotnej pamięci EEPROM sterownika i automatycznie odczytana po włączeniu zasilania.
0xFF	Test	Nie dotyczy	Funkcja testowa – zapalenie wszystkich segmentów wszystkich modułów LED z wykasowaniem zawartości bufora zmiennych (dla poszczególnych cyfr i kropek), bez wyczyszczenia pamięci EEPROM.

datkowo, jak wspomniano wcześniej, sterownik odpowiada na zaadresowanie go adresem wywołania ogólnego (0x00), co może być bardzo użyteczne w przypadku współistnienia wielu takich modułów na jednej magistrali I<sup>2</sup>C. W takim przypadku, za pomocą jednego rozkazu możemy zaadresować oraz sterować pracą wielu modułów

Tab. 2. Informacja o przyczynie błędu sterownika	
Informacja na wyświetlaczu	Interpretacja błędu
Err. 1	Błąd ramki transmisji – niewystarczająca liczba danych
Err. 2	Nieznany numer funkcji
Err. 3	Zbyt duża wartość Word (>9999)

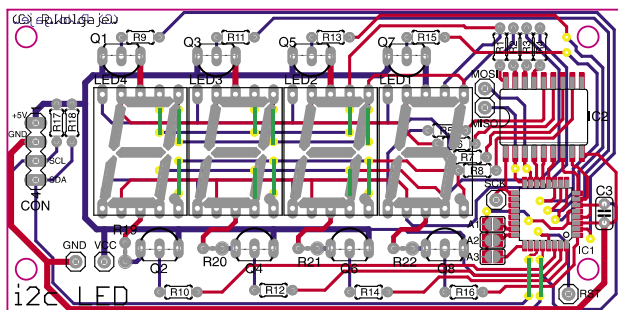
LED w tym samym czasie, mając na uwadze fakt, iż będą one realizować te same zadania. Przyjęta ramka transmisji ma następującą konstrukcję:  
Start – Adres układu <sup>ACK</sup> – Nr rozkazu <sup>ACK</sup> – Dana(1) <sup>ACK</sup> – Dana(x) <sup>ACK</sup> – Stop  
gdzie:  
Start – predefiniowany sygnał Start magistrali I<sup>2</sup>C,  
Adres układu – sprzętowy adres układu sterownika lub adres wywołania ogólnego,  
Nr rozkazu – żądany numer rozkazu niniejszego sterownika LED,  
Dana – dane związane z żądanym rozkazem w liczbie zależnej od właściwości rozkazu,

Stop – predefiniowany sygnał Stop magistrali I<sup>2</sup>C,  
Ack – sygnał potwierdzenia odbiornika

Dostępna lista rozkazów wraz z opisem ich właściwości oraz znaczeniem poszczególnych zmiennych przedstawiono w tab. 1.

W przypadku błędów transmisji oraz błędów po stronie programu użytkownika, sterownik wyświetli informację o przyczynie błędu podając jego numer. Interpretację informacji podano w **tab. 2**.

Należy także wspomnieć, iż na płycie naszego sterownika, która wymiarami zbliżona jest do standardowego modułu wyświetlacza LCD 2x16



Rys. 2. Schemat montażowy układu

znaków oraz posiada ten sam rozstaw otworów montażowych, przewidziano wyprowadzenia interfejsu SPI niezbędne dla zaprogramowania mikrokontrolera (program obsługi i bity konfiguracyjne (*fuse bits*)). Ma to fundamentalne znaczenie, gdyż, z uwagi na chęć minimalizacji obwodu drukowanego sterownika, zastosowano mikrokontroler w obudowie przeznaczony do montażu powierzchniowego, czyli takiej, jaka nie jest akceptowana przez większość programatorów.

### Montaż

Montaż układu należy rozpocząć od wlutowania elementów biernych (w tym rezystorów, które znajdują się

magistrali i zasilania oznaczone jako CON). Oczywiście złącze interfejsu SPI montujemy wyłącznie wtedy, gdy nie dysponujemy zaprogramowanym mikrokontrolerem. Główne złącze sterownika (CON) najlepiej zamontować od strony wyprowadzeń, gdyż pozwoli to na łatwe wpinanie modułu do odpowiedniego gniazda większych sterowników. Kłopotliwe może być lutowanie układów scalonych przeznaczonych do montażu powierzchniowego, zwłaszcza mikrokontrolera ATmega8, a to z uwagi na bardzo małe odległości pomiędzy poszczególnymi wyprowadzeniami. Najlepiej użyć do tego specjalnej stacji lutowniczej oraz odpowiednich topników lub też typowej

pod wlutowanymi później modułami LED), następnie montujemy tranzystory, układy scalone (montaż powierzchniowy), a na końcu moduły wyświetlaczy LED i złącza (pojedyncze piny dla interfejsu SPI i złącze kołkowe typu goldpin 1x4 dla

### WYKAZ ELEMENTÓW

#### Rezystory

R1...R8: 100 Ω/0,125 W  
R9...R16: 2,2 kΩ/0,125 W  
R17...R18: 4,7 kΩ/0,125 W  
R19...R22: 10 Ω/0,125 W

Rezystory R19...R22 są elementami opcjonalnymi niezbędnymi w przypadku zbyt małej różnicy (zdaniem użytkownika) pomiędzy czerwonym a pomarańczowym kolorem świecenia modułów LED. Elementy te są montowane pionowo

#### Kondensatory

C1: 100 nF ceramiczny

#### Półprzewodniki

IC1: ATmega8 (TQFP32)

Ustawienie bitów konfiguracyjnych: wewnętrzny oscylator RC 8 MHz (CKSEL: 0100, SUT: 10)

IC2: ULN2803 (SO18)

T1...T8: BC560

LED4...LED1: wyświetlacz LED TOS5121

#### Inne

CON: złącze goldpin 1x4

Vcc, GND, RST, SCK, MISO, MOSI: złącze goldpin 1x1

stacji, dobrej jakości pasty lutowniczej oraz taśmy rozlutowniczej. Poprawnie zmontowany układ nie wymaga jakiegokolwiek kalibracji i powinien działać bezpośrednio po włączeniu zasilania.

**Robert Wolgajew, EP**

**robert.wolgajew@ep.com.pl**

R E K L A M A

# RealView<sup>®</sup>

## oprogramowanie od ARMa dla ARMa



**KEIL<sup>™</sup>**  
An ARM<sup>®</sup> Company

**ARM<sup>®</sup>**

AUTORYZOWANY DYSTRYBUTOR

**KEIL<sup>™</sup>**  
An ARM<sup>®</sup> Company

**IAR SYSTEMS**

**CMX SYSTEMS**

**PHYTEC**

**LUMINARY MICRO**

**LUMINARY MICRO**

**WG**

Electronics

WG Electronics Sp. z o.o.

ul. Modzelewskiego 35

02-679 Warszawa

tel. +48 22 847 97 20

[www.wg.com.pl](http://www.wg.com.pl)