



Sterownik wyświetlacza LCD z interfejsem szeregowym

Typowo, moduł wyświetlacza LCD jest sterowany za pomocą interfejsu równoległego. W tym celu używa się 4 lub 8 bitów danych i 2 linie sterujące. Daje to razem 10 linii I/O, które nie zawsze są dostępne w mikrokontrolerze lub na płycie mikrokomputera. Prezentowany moduł umożliwia dołączenie typowego wyświetlacza tekstowego LCD o rozdzielczości 2 linie \times 16 znaków do urządzeń dysponujących tylko interfejsem szeregowym.

Rekomendacje: interfejs doskonale sprawdzi się z Arduino, Raspberry Pi i podobnymi mikrokomputerami, a także z „dużymi” komputerami PC.

Moduł interfejsowy pozwala na sterowanie wyświetlaczem alfanumerycznym ze sterownikiem HD44780 lub kompatybilnym. Doskonale sprawdzi się z popularnymi wyświetlaczami o rozdzielczości 2×16 znaków. Maksymalna obsługiwana rozdzielczość wyświetlacza wynosi 4×40 znaków. Sterowanie pracą urządzenia oraz przesyłanie treści do wyświetlenia odbywa się poprzez interfejs szeregowy, który sprzętowo może być zrealizowany na jeden z trzech sposobów: RS485, USB, UART. Zalety, których dostarcza takie rozwiązanie to przede wszystkim uproszczenie budowy urządzenia. Sam wyświetlacz wymaga połączenia z co najmniej 6 liniami I/O systemu nadrzędnego, a dodatkowo wymaga ustawiania kontrastu i sterowania podświetlaniem. Prezentowane urządzenie pozwala na rozbudowę dowolnego systemu o wyświetlacz kosztem tylko 2 lub nawet

1 linii I/O. Kolejną korzyścią jest uproszczenie programowe – system nadrzędny nie jest obciążony procedurami sterującymi wyświetlaczem – nie jest potrzebna dodatkowa biblioteka funkcji. Musi tylko mieć interfejs szeregowy, w który niejako domyślnie są wyposażone wszystkie mikrokomputery. Dodatkowo, mogą nim być przesyłane również komunikaty dla innych urządzeń, ponieważ moduł reaguje tylko na komendy w ściśle określonym formacie.

Sposoby sterowania

Moduł może być sterowany za pomocą jednego z dwóch protokołów komunikacyjnych: nieskomplikowanych komend tekstowych lub zbliżonego do Modbus RTU.

W pierwszym komendy sterujące urządzeniem mogą być wysyłane nawet z klawiatury przez dowolny program typu terminal

DODATKOWE MATERIAŁY NA FTP:

<ftp://ep.com.pl>

USER: 95777, PASS: 53wtjyf6

W ofercie AVT*

AVT-5615

Podstawowe informacje:

- Interfejs USB, UART, lub RS485.
- Nieskomplikowane komendy sterujące lub protokół Modbus RTU.
- Możliwość komunikacji za pomocą dwóch lub tylko jednego przewodu.
- Obsługuje wyświetlacze LCD tekstowe, zgodne z HD44780, o rozdzielczości od 1×6 do 4×40 .
- Szeroki zakres napięcia zasilania 6...30 V DC. Maksymalny pobór prądu 200 mA.
- Wymiary 80 mm \times 37 mm (wielkość zbliżona do wyświetlacza 2 linie \times 16 znaków).

Projekt pokrewne na FTP:

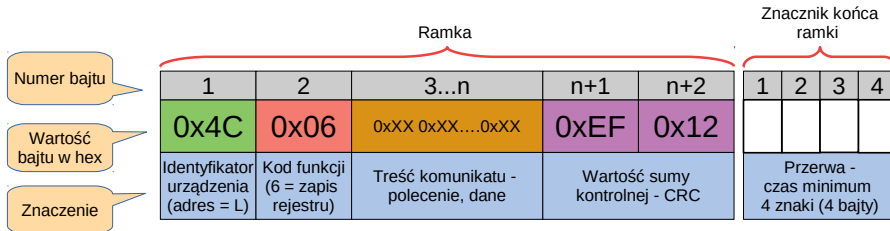
(wymienione artykuły są w całości dostępne na FTP)

---	Sterownik graficznego wyświetlacza LCD 8,4" typu EG9018C (EP 11/2013)
AVT-1737	LCD Expander dla Arduino (EP 4/2013)
AVT-1715	AVTduinoGraphLCD (EP 11/2012)
AVT-1615	AVTduino LCD. Wyświetlacz LCD dla Arduino (EP 4/2011)
---	Obsługa graficznych wyświetlaczy TFT z chipsetem FSA506 (EP 6/2010)

* **Uwaga!** Elektroniczne zestawy do samodzielnego montażu. Wymagana umiejętność lutowni! Podstawową wersją zestawu jest wersja [B] nazywana potocznie KItem (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wylutować 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 wylutowane w płytkę PCB)
- wersja [A] płytką drukowaną bez elementów i dokumentacją
- Kity w których występuje układ scalony wymagający zaprogramowania, posiadają następujące dodatkowe wersje:
 - wersja [A+] płytką 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>

**Rysunek 1. Ogólny format komend tekstowych**

tekstowy. Ogólny format komend pokazano na **rysunku 1**. Pierwszy znak ramki to identyfikator urządzenia – adres identyfikujący moduł, do którego jest kierowana komenda. Domyślny identyfikator to duża litera „L”. Identyfikator można zmienić, ustawić indywidualny dla każdego urządzenia. Aby zakończyć komendę należy wysłać znak „CR” (\r), a następnie utrzymać „ciszę” na magistrali przez czas dłuższy niż potrzebny na transmisję 4 znaków (podobnie jak w protokole Modbus). Komenda może sterować różnymi funkcjami. Ich wykaz wraz z opisem umieszczono w **tabeli 1**. Uwaga! W przykładach komend dla przejrzystości wstawiono dodatkowe znaki odstępu. W rzeczywistej sytuacji te znaki są ignorowane.

Spacje mogą występować tylko w treści wysłanej na ekran wyświetlacza.

Każda prawidłowo odebrana komenda zostanie potwierdzona komunikatem „OK” zakończonym znakiem nowej linii „CR”. Przykładowy kod programu dla Arduino konfiguruje port szeregowy i wypisujący tekst „Hello” na wyświetlaczu (dołączonym poprzez prezentowane urządzenie) wygląda następująco:

```
Serial.begin(9600);
```

```
Serial.print(„LWHello\r”);
```

Drugi protokół obsługiwany przez urządzenie jest wzorowany na protokole Modbus RTU, ale nie ma jego pełnej funkcjonalności. Niemniej, w zupełności wystarczy do tego, aby sterować wyświetlaczem za pomocą

sterownika PLC wyposażonego w interfejs RS485. Ogólną strukturę komend pokazano na **rysunku 2**. Każda ramka rozpoczyna się identyfikatorem urządzenia, zawiera funkcję do wykonania, dane związane z tą funkcją i jest zakończona dwoma bajtami sumy kontrolnej będącymi głównym elementem mechanizmu kontroli poprawności przesyłanych danych.

Moduł sterujący wyświetlaczem obsługuje dwie funkcje z palety funkcji Modbus: zapis pojedynczego rejestru oraz zapis wielu rejestrów. Dokładną strukturę komend z tymi funkcjami oraz odpowiedzi urządzenia pokazano na **rysunkach 3 i 4**, natomiast w **tabeli 2** umieszczono opis przeznaczenia poszczególnych rejestrów.

W standardzie Modbus RTU przyjęto założenie, że operacje na rejestrach dotyczą rejestrów mieszczących 16 bitów. Pomimo tego, że moduł sterujący wyświetlaczem ma rejestry 8 bitowe, to dane przesyłane w komendach interpretuje jako liczby 16-bitowe, a następnie ogranicza je do 8 bitów.

Do modułu można na dwa sposoby przesyłać tekst do wyświetlenia. Pierwszy z nich polega na przesłaniu tekstu po dwa znaki, jako jeden rejestr 16-bitowy. Należy wtedy zadbać o prawidłowe przygotowanie takich danych oraz prawidłowe ich odczytanie. Poza tym, pozostaje problem tekstu o nieparzystej liczbie znaków. Drugie rozwiązanie to przesyłanie po jednym znaku dla każdego rejestru. Wtedy starszy bajt rejestru zawsze będzie zerem i w rezultacie przesyłane będą puste dane. W module zastosowano rozwiązanie, które sprawdzi się niezależnie od tego, jaką metodą przesyłania tekstu zastosujemy. Oprogramowanie sterownika odczytuje dane jako bajty, więc może odczytać dwa znaki z rejestru 16-bitowego, ale pomija bajty będące zerem. Dzięki temu, w wypadku pierwszej metody, ostatni znak w tekście o nieparzystej liczbie znaków może wynosić 0 i nie zostanie wyświetlony a pozostałe tak. W drugim wypadku, pomimo tego, że każdy bajt będzie oddzielony od poprzedniego bajtem zerowym, to na wyświetlaczu zostaną pokazane obok siebie.

REKLAMA

Projekty na...Texas

STM32

www.stm32.eu

KAMAMI

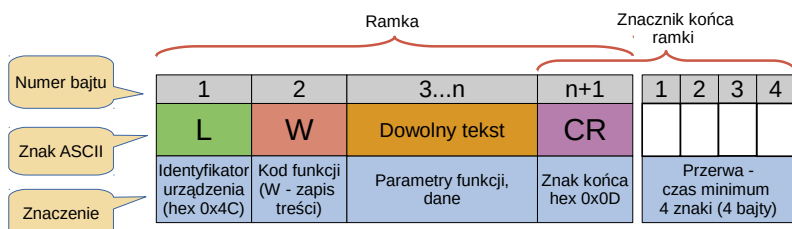
Tabela 1. Wykaz i opis funkcji protokołu komend tekstowych

Kod funkcji (nazwa)	Działanie funkcji	Przykład komendy	Uwagi
W (Write)	Dodaje przestaną treść do treści na wyświetlaczu.	L W dodana treść \r	Rozpoczyna w miejscu ostatniego położenia kursora.
D (Delete)	Usuwa wyświetlaną treść.	L D \r	Ustawia kursor na pozycji początkowej.
P (Position)	Ustawia kursor na wybranej pozycji. Wymaga podania dwóch parametrów: numeru wiersza i numeru znaku. Parametry muszą być oddzielone znakiem przecinka.	L P 0, 6 \r	Wartości liczone są od zera. Przykład ustawia kursor w pierwszym wierszu na siódmym znaku.
B (Backlight)	Ustawia intensywność podświetlania wyświetlacza w zakresie 0...255.	L B 128 \r	Ustawia intensywność podświetlania na wartość 128, czyli 50% zakresu.
X (Extreme)	Wykonuje sprzętowy restart wyświetlacza. Przydatne w sytuacji gdy na ekranie pokazują się „krzaczkę”.	L X \r	Nie usuwa wyświetlanej treści, przywraca treść na ekran po wykonaniu restartu.
C (Config)	Ustawia konfigurację sterownika - osiem parametrów: baudrate, identyfikator, liczbę wierszy, liczbę kolumn, organizację, kontrast, współ. temperatury, podświetlenie. Dokładny opis w tabeli 3.	L C 2,L,2,16,0,40,1,200 \r	Wartości parametrów zostaną automatycznie zapisane w pamięci nieulotnej i będą stosowane przy każdym kolejnym uruchomieniu sterownika.

Pozycja kursora, czyli miejsce na ekranie, w którym zostanie wyświetlony przesłany tekst jest określana numerem rejestru, do którego odnosi się komenda (dotyczy

rejestrów o numerach 0...159). Pozycja kursora jest liczona od 0, więc funkcja zapisu rejestru o numerze np. 9 ustawi pierwszy znak na pozycji 10 w pierwszej linii wyświetlacza

o rozdzielczości 2×16. Zapis do rejestru o numerze np. 18 ustawi pierwszy znak na pozycji 3 w drugiej linii wyświetlacza o rozdzielczości 2×16. Kolejne znaki tej komendy będą umieszczane na wyświetlaczu jeden za drugim, niezależnie od tego, do którego rejestru byłyby przypisane.



Rysunek 2. Ogólna postać komend protokołu zbliżonego do Modbus RTU

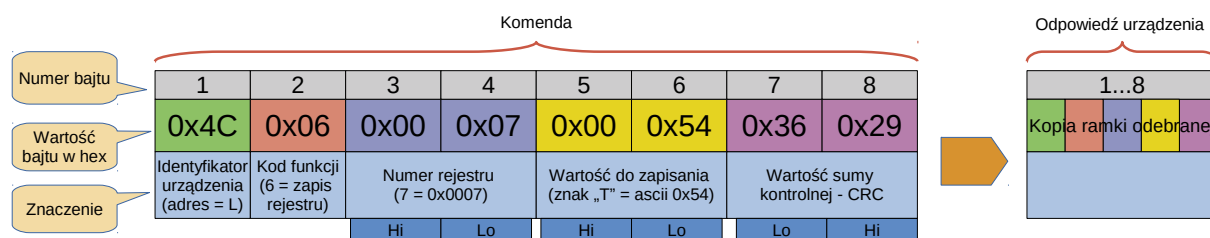
Tabela 2. Opis rejestrów urządzenia			
Numer rejestru	Zakres wartości	Znaczenie, funkcja	Uwagi
0...159	Znaki ASCII o wartości większej od 0	Zawartość tego obszaru przenoszona jest na ekran wyświetlacza.	Numer pierwszego rejestru oznacza miejsce umieszczenia pierwszego znaku
160	Znaki ASCII	Dodaje jeden znak w ostatnim położeniu kursora	Uruchamiane tylko poprzez funkcję zapisu jednego rejestru (funkcja o wartości 6)
161	1	Powoduje wyczyszczenie treści wyświetlacza	
162	1	Wykonuje restart wyświetlacza (bez utraty treści)	
200...207	Znaki ASCII o wartości większej od 0	Osiem rejestrów konfiguracyjnych	Po przestaniu kompletu parametrów wartości zostają zapamiętane w pamięci nieulotnej

Rejestry konfiguracyjne

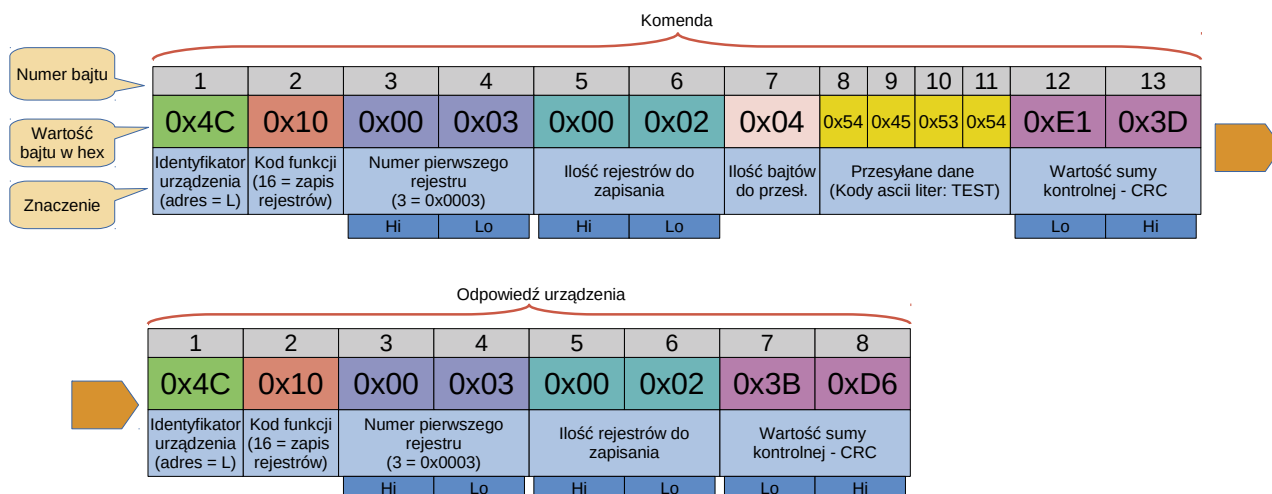
Służą do ustawienia podstawowych parametrów modułu. Dostęp do nich jest możliwy poprzez funkcję o kodzie „C” w protokole komend tekstowych, a w protokole Modbus poprzez funkcje zapisu grupy rejestrów zaczynających się od rejestru o numerze 200. **Uwaga!** Moduł pozwala na przesłanie tylko pełnej grupy rejestrów konfiguracyjnych, czyli tylko 8 rejestrów zaczynając od rejestru o adresie 200. Przesłane ustawienia zostają od razu zastosowane i zapamiętane w pamięci nieulotnej. Dotyczy to także prędkości transmisji oraz identyfikatora urządzenia. W protokole Modbus można zmienić wartość jednego rejestru spośród rejestrów konfiguracyjnych, ale takie ustawienie nie zostanie zapisane w pamięci nieulotnej i po ponownym uruchomieniu zostanie przywrócona poprzednia wartość. Funkcje i opis rejestrów konfiguracyjnych umieszczono w tabeli 3.

Budowa i montaż

Schemat ideowy modułu sterującego wyświetlaczem pokazano na rysunku 5. Blok zasilania zbudowano z użyciem układu



Rysunek 3. Struktura komendy z funkcją zapisu jednego rejestru



Rysunek 4. Struktura komendy z funkcją zapisu wielu rejestrów

Tabela 3. Opis rejestrów konfiguracyjnych

Numer parametru	Zakres wartości	Funkcja	Uwagi
1	0...7	Określa prędkość transmisji – baudrate [bps]. 0 = 2400, 1 = 4800, 2 = 9600, 3 = 14400, 4 = 19200, 5 = 28800, 6 = 57600, 7 = 115200.	Wartość domyślna wynosi 2 czyli prędkość 9600 bps.
2	1...247	Identyfikator urządzenia – adres. Wartość (znak), którą muszą rozpoczynać się komunikaty kierowane do urządzenia.	Domyślna wartość wynosi 76 co odpowiada znakowi L w tablicy ascii.
3	1...4	Odnosi się do budowy zastosowanego wyświetlacza, określa ilość wierszy na ekranie.	Domyślna wartość wynosi 2 co odpowiada wyświetlaczowi o organizacji np. 2×16 znaków.
4	1...40	Odnosi się do budowy zastosowanego wyświetlacza, określa ilość znaków w jednym wierszu na ekranie.	Domyślna wartość wynosi 16 co odpowiada wyświetlaczowi o organizacji np. 2×16 znaków.
5	0...1	Odnosi się do budowy zastosowanego wyświetlacza, określa sposób organizacji pamięci obrazu. Wyświetlacz 4×16 może mieć pamięć zorganizowaną w konfiguracji 2×32. Aby treść była wyświetlana w prawidłowej kolejności należy wprowadzić odpowiednie korekty.	Wartość domyślna wynosi 0 czyli korekta wyłączona.
6	0...255	Wartość kontrastu. Urządzenie określa intensywność kontrastu podając odpowiedni sygnał (napięcie) do wyświetlacza. Poziom tego napięcia można określić wartością tego rejestru, zakres wynosi 0...5 V.	Domyślna wartość wynosi 40. Przy zbyt wysokiej wartości nie będzie widoczna treść, przy zbyt niskiej cały ekran będzie włączony (tzw. kwadraciki).
7	1...10	Kontrast wyświetlacza zmienia się wraz z temperaturą. Aby utrzymać optymalną ostrość obrazu należy korygować kontrast wraz ze zmianą temperatury.	Domyślna wartość wynosi 1. Wartość 0 oznacza wyłączoną funkcję korekty kontrastu.
8	0...255	Określa jasność podświetlania wyświetlacza. Przy wartości maksymalnej obraz jest najbardziej widoczny. Ustawienie wartości mniejszej pozwala zmniejszyć pobór energii.	Domyślna wartość wynosi 200.

Wykaz elementów:**Rezystory:** (SMD 1206)

R1, R2, R4, R8, R10, R12, R17, R18: 4,7 kΩ

R3: 1,5 kΩ

R5, R6, R9, R11, R15: 10 Ω

R7: 1 Ω

R13, R14, R16: 100 Ω

Kondensatory:

C1: 220 μF/35 V

C2: 100 μF/6,3 V (SMD „B”)

C3...C6, C8, C9, C12, C13: 100 nF (SMD 1206)

C7: 100 pF (SMD 1206)

C10, C11, C14, C15: 10 pF (SMD 1206)

C16: 1 μF (SMD 1206)

Półprzewodniki:

D1...D3: SS14

LED1: LED SMD 1206

T1: BC847

IC1: L5973

IC3: FT230XS

IC4: Atmega168 (SMD)

IC6: MCP2551

IC7: LM50

Inne:

X1: 11,0592 MHz

L1: dławik 15...47 μH/0,5 A (SMD)

IC2: gniazdo goldpin 2×8

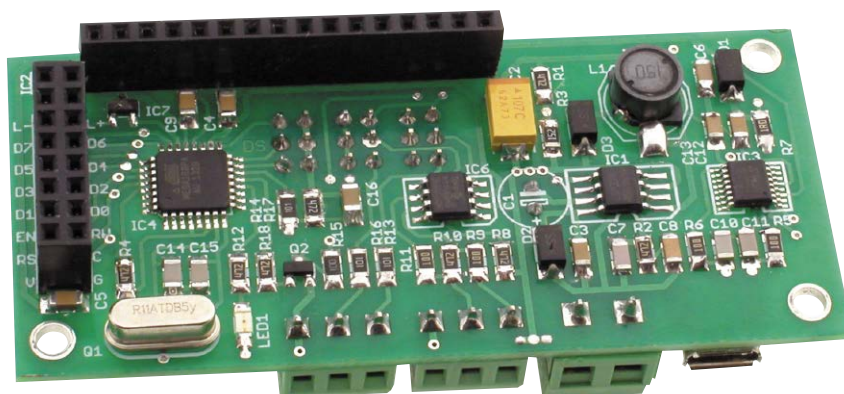
IC5: gniazdo goldpin 1×16

CFG1...CFG3: goldpin 2×3 + zworka

CON1: DG301-5/2

CON2, CON3: DG381-3,5/3

CON4: micro USB



napięcie stabilizowane 5 V – może ono być doprowadzone również ze złącza USB, ale ze względu na spadek napięcia na diodzie D1, jego wartość będzie nieco niższa.

Układ IC3 typu FT230 pracuje w typowej aplikacji, w której pełni rolę konwertera USB/UART. Układ IC6 (MCP2551) jest wykorzystywany niestandardowo – jest to driver CAN, ale w prezentowanym urządzeniu jest driverem magistrali RS485 i doskonale sprawdza się w tej roli. Przewyższa typowe rozwiązania np. z MAX485 odpornością na uszkodzenia, ale „przegrywa”, jeśli chodzi o maksymalną długość przewodu połączeniowego. Złącza szpilkowe CFG2 i CFG3 pozwalają na wybranie magistralę do komunikacji, przełączając odpowiednie sygnały do portu szeregowego UART mikrokontrolera.

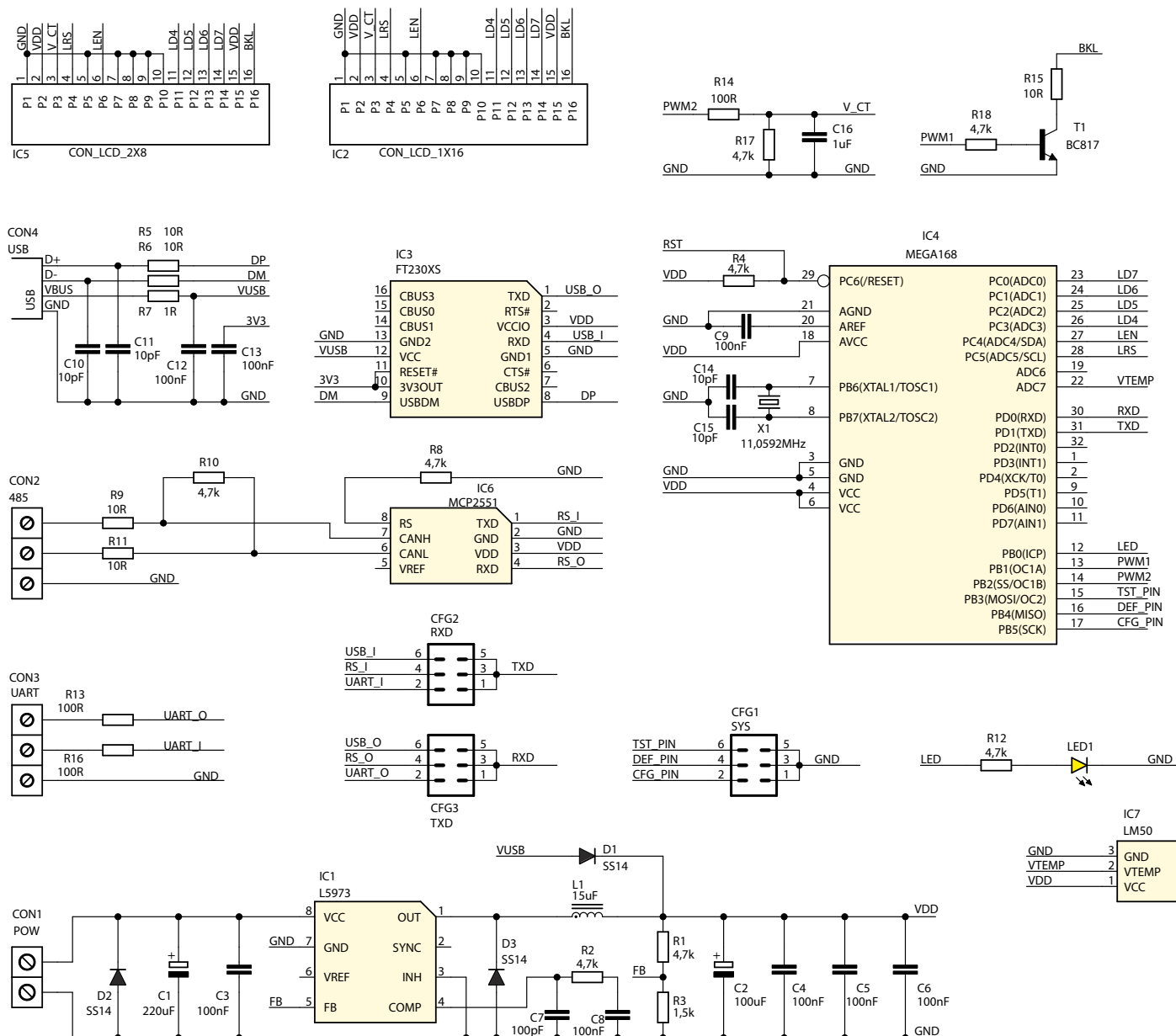
Złącza IC2 oraz IC5 służą do przyłączenia wyświetlacza. Zastosowano dwa typy złącz dla dwóch najpopularniejszych konfiguracji wyprowadzeń wyświetlaczy. Informacji o temperaturze dostarcza układ LM50 i dzięki temu jest możliwa korekcja kontrastu. Napięcie sterujące kontrastem jest wytwarzane za pomocą wbudowanego w mikrokontroler generatora PWM z wyjściowym filtrem RC.

Schemat montażowy modułu pokazano na **rysunku 6**. W czasie montażu należy pamiętać o tym, aby złącza CON1...CON4, szpilki konfiguracyjne CFG1...CFG3 oraz kondensator C1 zamontować od spodu płytki, jak na **fotografii 7**.

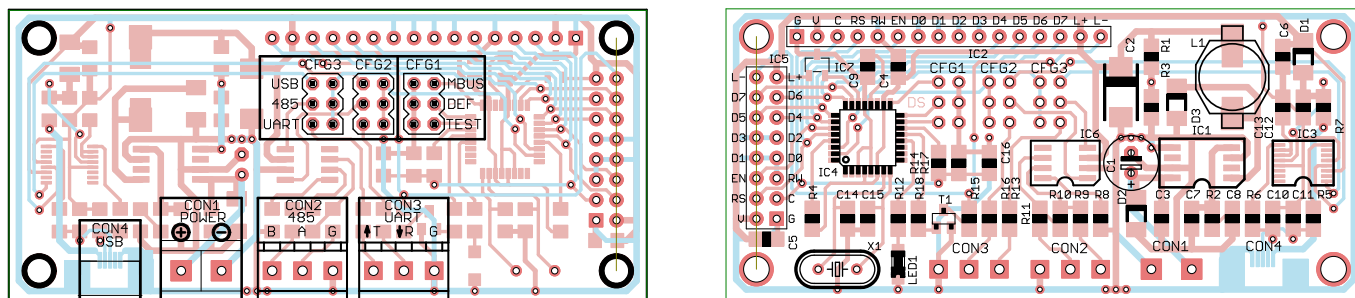
Uruchomienie i obsługa

Zmontowane urządzenie, z dołączonym wyświetlaczem, można łatwo sprawdzić bez

scalonego L5973D impulsowej przetwornicy obniżającej napięcie. Dzięki dużej sprawności jest możliwe zasilanie urządzenia napięciem aż do 30 V. Na wyjściu otrzymujemy

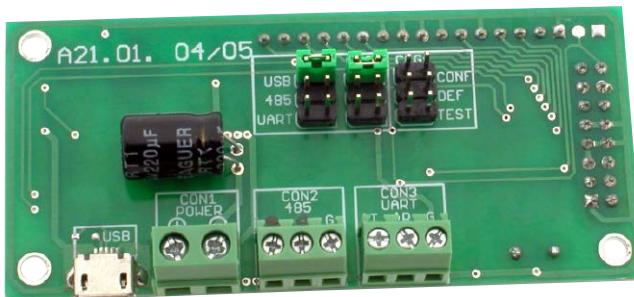


Rysunek 5. Schemat ideowy modułu sterującego wyświetlaczem

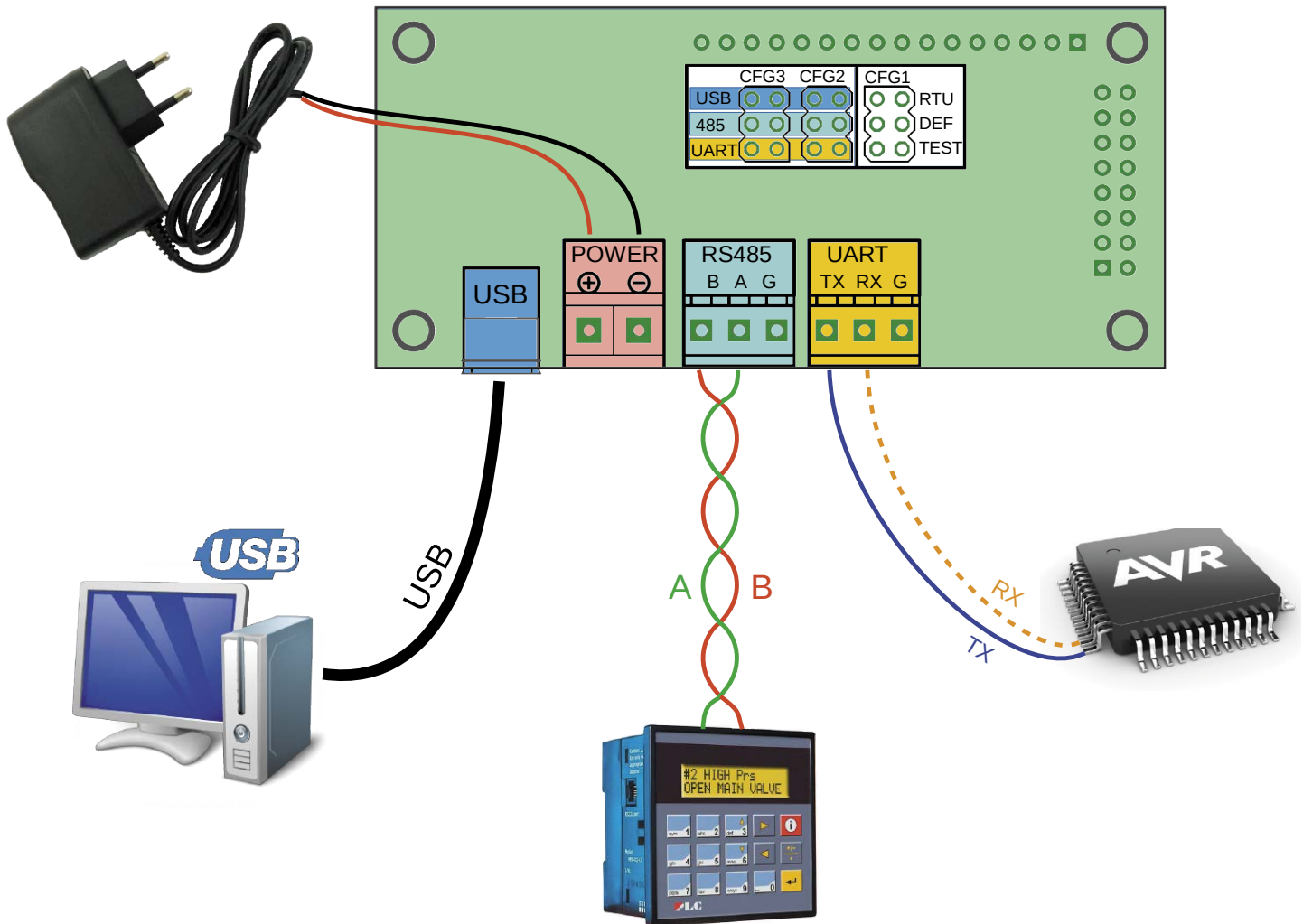


Rysunek 6. Schemat montażowy modułu sterującego wyświetlaczem

wysyłania jakiejkolwiek komendy. Po włączeniu zasilania powinno włączyć się podświetlenie wyświetlacza, ponieważ domyślnie jest ono ustawione na ok. 80% intensywności świecenia. Następnie należy założyć zwózkę na szpilkach oznaczonych CFG1 na pozycji TEST. W tym trybie urządzenie wypełnia wyświetlacz ciągiem liter a następnie zmienia poziom kontrastu od wartości minimalnej do wartości maksymalnej w pętli



Fotografia 7. Montażu elementów od spodu płytki drukowanej



Rysunek 8. Połączenie interfejsu transmisyjnego i odpowiadająca im konfiguracja zwrotek

nieskończonej. Dzięki temu, w każdym cyklu pętli będzie moment, gdzie kontrast będzie optymalny dla wyświetlacza i wyświetlana będzie prawidłowa treść. Warto w takim momencie zdjąć zworkę ze szpilek TEST, ponieważ wtedy oprogramowanie modułu zapamięta aktualną wartość kontrastu. Oczywiście nie jest to procedura konieczna, kontrast można ustawić także odpowiednimi komendami.

Szpilki „DEF” pozwalają na przywrócenie ustawień domyślnych – wystarczy w dowolnej chwili założyć zworkę. Szpilki oznaczone RTU służą do wyboru protokołu komunikacyjnego, brak zworki oznacza protokół komend tekstowych, a zworka założona to protokół Modbus RTU.

Szpilki oznaczone CFG2 oraz CFG3 służą do wyboru magistrali komunikacyjnej (nie protokołu) i muszą być ustawiane obie w tym

samym położeniu. Urządzenie może obsługiwać tylko jedną magistralę w danej chwili. Sposób połączenia różnych magistrali oraz ustawienie zwrotek przedstawia **rysunek 8**. W wypadku UART połączenie może być wykonane tylko jednym przewodem – wystarczy sygnał TX z mikrokomputera do modułu oraz jest konieczne, aby połączone urządzenia miały wspólną masę zasilania.

KS

REKLAMA

Prenumerujesz
„Elektronikę Praktyczną”
i „Elektronikę dla Wszystkich”?
Masz prawo do
bezpłatnej prenumeraty
miesięcznika „Elektronik”
w promocji 1+1=3

www.avt.pl/prenumerata

