



WKS KP PANEL
Panel sterujący z interfejsem RS485
38400 b/s 8N1

Instrukcja obsługi



RoHS

Producent: EL KOSMITO Rafał Majewski
Ul. Kościuszki 21
68-320 Jasień
NIP 928-192-12-96
REGON 080936699

Kontakt: www.elkosmito.pl
info@elkosmito.p

Spis treści

Opis ogólny.....	3
Cechy modułu WSKKP PANEL.....	3
Parametry techniczne.....	3
Podłączenie modułu.....	4
Komunikacja.....	4
Dodatek A: Instalacja sieci RS485.....	12
Uwagi.....	13

Opis ogólny.

Firma **EL KOSMITO** opracowała panel sterujący wyposażony w interfejs RS485, dzięki czemu jest możliwość pobierania danych z panelu nawet na dużych odległościach. Można np. sterownik główny umieścić w szafie sterującej a panel na ścianie w innym pomieszczeniu. Oczywiście nie stoi nic na przeszkodzie aby wykorzystać go do komunikacji z komputerem. Panel jest wyposażony w:

- 18 wyświetlaczy 7-segmentowych zgrupowanych po 3 do wyświetlania liczb (szczegóły i zasada w dalszej części instrukcji)
- 6 potencjometrów z przetwornikiem analogowo-cyfrowym 10 bitowym
- 5 przycisków sterujących
- wyświetlacz LCD alfanumeryczny 16x2 znaki z podświetleniem (podświetlenie zawsze włączone)
- diody sygnalizujące pracę interfejsu RS485

Sterownik posiada wbudowany kontroler wyświetlacza LCD, dzięki któremu obsługa całego panelu jest dość prosta. Nawet obsługa wyświetlacza LCD nie jest skomplikowana. Zastosowanie gotowych funkcji daje możliwość wyświetlania 4 pól tekstowych i maksymalnie 4 napisów przesuwanych. Dodatkowo oczywiście na wyświetlaczu możemy dodawać własne znaki, korzystać z bazy niektórych znaków specjalnych, krótko mówią obsługa LCD jest banalna w stosunku do obsługi bezpośredniej.

Tak samo łatwo jest ustawiać wartości liczbowe na wyświetlaczach 7-segmentowych. Nie trzeba budować znaków z siedmiu segmentów, a po prostu przysyłać wartości liczbowe. Układ prawidłowo je interpretuje i wyświetli tyle co się da, czyli:

- dla liczb od -99 do -10 wyświetla wartości poprzedzone znakiem - (minus) oraz dwie cyfry, bez części ułamkowej
- dla liczb od -9.9 do -0.1 wyświetla wartości poprzedzone znakiem - (minus) oraz jedną cyfrę jedności i jedną cyfrę części ułamkowej (a więc -5.72 wyświetli się jako -5.7, a -6 wyświetli się jako -6.0)
- dla liczb od 0 do 99.9 wyświetla się wartość cyfr dziesiątek i jedności oraz jedna cyfra części ułamkowej (a więc 4.6 wyświetli się jako 4.6 bez zera na początku, czyli nie 04.6, a wartość 56.93 wyświetli się jako 56.9)
- dla liczb od 100 do 999 wyświetlają się 3 cyfry części całkowitej, bez części ułamkowej

Dodatkowo wyświetlacze 7-segmentowej posiadają możliwość wyświetlania liczb w trybie ciągłym lub migającym. A więc wartości na wyświetlaczu mogą cały czas świecić lub migać z częstotliwością około 1Hz

Stan potencjometrów odczytać jest też bardzo łatwo i o tym przekonamy się w dalszej części instrukcji.

Przy odczytywaniu przycisków mamy dostęp do wykrywania impulsów (dzięki czemu łatwo zapanować nad rozróżnianiem impulsu od ciągłego trzymania) i ciągłego przytrzymania z licznikiem sekund kiedy przycisk jest trzymany (daje to możliwość realizacji funkcji dłuższego przytrzymania i szybszego przewijania wartości podczas ustawiania itp. przez końcowego użytkownika).

Krótko mówiąc: spore możliwości w jednym urządzeniu.

Cechy modułu WKSKP PANEL

- ✓ Komunikacja RS485
- ✓ 6 potencjometrów z przetwornikiem analogowo-cyfrowym 10-bitowym
- ✓ 18 wyświetlaczy 7-segmentowych o wysokości 14,2mm (6 liczb po 3 cyfry)
- ✓ 5 przycisków sterujących
- ✓ wyświetlacz alfanumeryczny 16x2 znaki oczywiście z wbudowanym kontrolerem do obsługi przez RS485
- ✓ Możliwość używania długich magistrali (nawet setki metrów)
- ✓ Do instalacji wystarczy przewód skrętka (np. komputerowa)
- ✓ Wymiary 260x125x60
- ✓ Możliwość montażu na ścianie
- ✓ Wykonanie IP20
- ✓ Zakres temperatur pracy modułu -10 do 45°C

Parametry techniczne

- ✓ Zasilanie: DC 7-28V 2A (w zależności od napięcia im mniej tym więcej 12V 2A / 24V 1A)
- ✓ Komunikacja: RS485 38400 b/s 8N1, wymagane przełączanie z nadawania na odbiór poniżej 0,3ms
- ✓ Zasięg interfejsu: kilkaset metrów
- ✓ Podłączenie: skrętka, impedancja 120 ohm
- ✓ Przetwornik A/C potencjometrów: 10-bit (wartości od 0 do 1023)

- ✓ Wyświetlacze 7-segmentowe: 14,2mm wysokość, kolor czerwony
- ✓ Wyświetlacz LCD: oparty na standardzie HD44780 16x2 znaki, podświetlenie zielone włączone
- ✓ Separacja RS485: brak, możliwość wykonania wersji specjalnych
- ✓ Tryb pracy modułu: SLAVE (sam nie nadaje bez pytania)
- ✓ Adresowanie modułu: brak, opcja dostępna dla wersji specjalnych
- ✓ Zgodność RoHS
- ✓ Wykonanie IP20
- ✓ Zakres temperatur pracy modułu -10 do 45°C

Moduł w sieci na jednej magistrali może być jeden. Na specjalne zamówienie można wykonać wersje z adresowaniem umożliwiające używanie kilku modułów na jednej magistrali.

Podłączenie modułu

Należy odkręcić śrubki z dołu i z góry obudowy. Nie odkręcaj wszystkich śrub! Odkręć tylko 4, te które są na połowie obudowy z potencjometrami, wyświetlaczami. Te na drugiej połowie pozostaw.

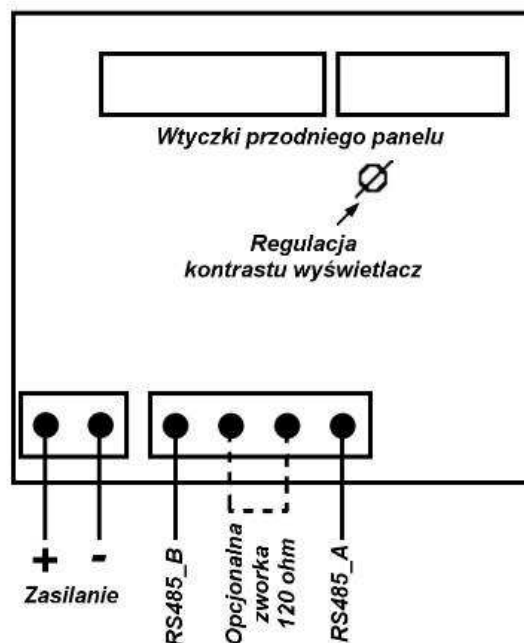
Następnie zdejmij obudowę i wyjmij 2 wtyczki, aby oddzielić przód panelu od tyłu. Teraz możesz powiesić obudowę i podłączyć układ zgodnie ze schematem na Rys. 1. Po podłączeniu włóż wtyczki, złoż obie części obudowy z bokami i skręć. Jak widać nie ma w podłączeniu niczego skomplikowanego.

Wyjaśnienia wymagać może jedynie opcjonalna zworka. Należy założyć zworkę jeśli moduł jest pierwszym lub ostatnim na linii magistrali (patrz dodatek A).

Z przodu obudowy znajdują się diody:

- żółta – pokazuje kiedy układ odbiera sygnał transmisji RS485
- zielona – pokazuje kiedy układ nadaje sygnał transmisji RS485

Po włączeniu zasilania na panelu wyświetla się sekwencja testowa na wyświetlaczach 7-segmentowych oraz napis powitalny na wyświetlaczu LCD.



Rys. 1: Podłączenie układu

Komunikacja

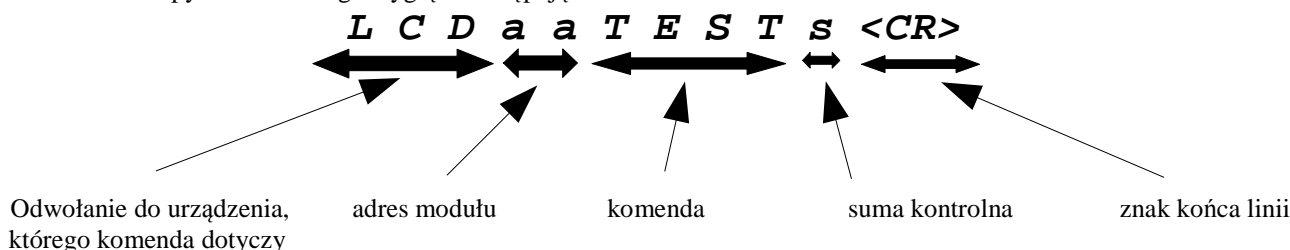
Układ pracuje z interfejsem RS485 skonfigurowanym na 38400 b/s 8N1. Warto tylko zadbać, aby w przypadku przejściówek RS232 na RS485 i podłączeniu do komputera, przejściówka zapewniała czas przełączania z nadawania na odbiór w czasie poniżej 0,3ms. Jest to o tyle ważne, że przejściówki pracujące od niskich prędkości mogą mieć na stałe ustawiony czas przejścia z nadawania na odbiór np. 1ms, podczas gdy przy wyższych prędkościach transmisji tak długi czas jest bez sensu. Moduł wymaga aby czas przełączenia pomiędzy nadawaniem i odbiorem był poniżej 0,3ms. W przypadku transmisji w układach z mikrokontrolerami ten problem praktycznie nie istnieje, gdyż przełączanie odbywa się programowo-sprzętowo, a nie tylko sprzętowo jak w przypadku przejściówek.

Do modułu można pobrać ze strony www.elkosmito.pl prosty program do testowania i sprawdzenia poprawnej pracy panelu. Aplikacja konsolowa pobiera i wypisuje wszystkie dane oraz steruje pracą wyświetlaczy 7-segmentowych..

Komunikację można także przetestować przy pomocy popularnych programów terminali RS232 takich jak Realterm, ale w tym przypadku jest to dość kłopotliwe, bo komendy mogą składać się z wielu znaków, więc lepiej realizować to w programach testowych.

Znamy już szczegóły sprzętowe interfejsu, a więc przejdźmy do szczegółów protokołu. Protokół komunikacji jest bardzo prosty i tekstowy. Moduł WSKP PANEL jest modułem typu SLAVE a więc sam nic nie wysyła jak nie zostanie zapytany. Nie powinny w związku z tym występować kolizje.

Format zapytania testowego wygląda następująco:



Format odpowiedzi na zapytanie testowe wygląda następująco:

O K <CR>

Na przykładzie zapytania testowego omówimy sobie kilka szczegółów. Zapytanie o treści „LCDaaTESTs” (ważna wielkość liter) nie stanowi problemów do zrozumienia. Jako adres wpisujemy 00 (chyba, że mamy wersję z możliwością ustawienia innego adresu). Potem jest nazwa komendy a następnie podajemy jeden znak sumy kontrolnej. Jak jest obliczany to za chwilę sobie wyjaśnimy. Ostatni znak to znak końca linii. Ważne, że za znak końca linii uważa się znak <CR> ('r') a nie znak <LF> ('n'). Po wysłaniu odpowiedniej treści, układ zwróci odpowiedź. Ważne i wymagające podkreślenia jest to, że układ nie pracuje w żadnym trybie ECHO, a więc nie po nadaniu znaku, nie wysyła go zwrótnie. W praktyce to żaden problem, bo wiele programów typu np. Realterm pozwalają na włączenie „lokalnego echo”, a na dodatek poza testami „echo” w ogóle nie jest nam potrzebne.

Przejdźmy sobie do sumy kontrolnej. Jej obliczenie jest dość proste. Jest to nic innego jak obliczenie sumy wartości znaków ascii poprzedzających znak sumy kontrolnej, a następnie obliczenie reszty z dzielenia przez 71 i do wyniku dodanie 48. Tak powstała wartość jest liczbą, która będzie jakimś znakiem z tablicy ASCII i właśnie ten znak jest znakiem sumy kontrolnej. Spójrzmy na przykład:

Oblicz sumę kontrolną dla:

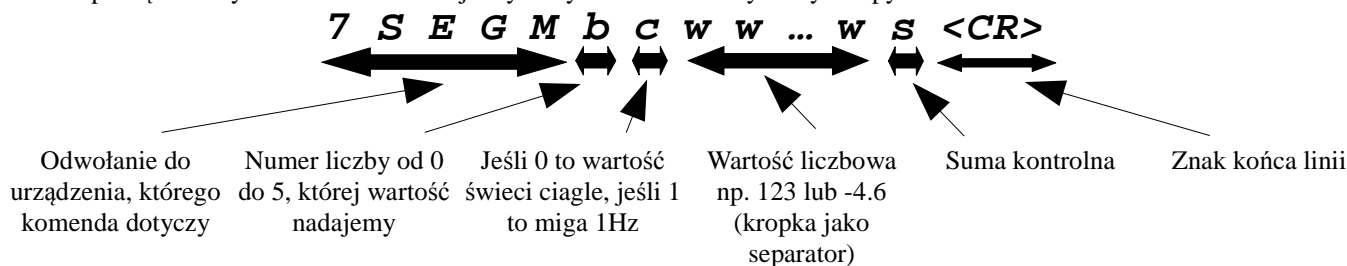
LCD00TEST

$$\begin{aligned} \text{Suma kontrolna} &= [('L' + 'C' + 'D' + '0' + '0' + 'T' + 'E' + 'S' + 'T') \bmod 71] + 48 = \\ &= [(76+67+68+48+48+84+69+83+84) \bmod 71] + 48 = \\ &= [627 \bmod 71] + 48 = \\ &= 59 + 48 = \\ &= 107 = \\ &= 'k' \end{aligned}$$

A więc chcąc nadać pełną komendę nadalibyśmy:

LCD00TESTk

Widać, że to nic skomplikowanego. Teraz najpierw zajmiemy się wyświetlaczami 7-segmentowymi, bo LCD to za dużo na początek. Aby ustawić wartość na jednym wyświetlaczu korzystamy z zapytania



Komenda zwraca:

O K <CR>

Przykład:

7SEGM40-23.73[

Przesłanie takiej komendy spowoduje wyświetlenie na 4 grupie wyświetlaczy (liczone są od lewej do prawej 0,1,2 potem drugi rząd 3,4,5, a więc na dolnych wyświetlaczach na środku) wartości -23 (czyli tyle znaków ile się zmieści).

Układ po odebraniu komendy zwraca OK, przy czym koniecznie trzeba zaznaczyć, że w większości przypadków OK nie oznacza, że komenda będzie prawidłowo sformułowana, tylko oznacza, że odebrana komenda ma poprawną sumę kontrolną i dotyczy ona wybranego urządzenia ('7SEGM'). A więc przesłanie:

7SEGM40-EsWm

również jest prawidłowe, gdyż zgadza się suma kontrolna, a więc zostanie zwrócone OK.

W przypadku kiedy panele wykonane są w specjalnej wersji z możliwością zmiany numeracji adresu to komenda

dotycząca wyświetlacza wyglądałaby następująco:

7 S E G M a a b c w w ... w s <CR>

czyli należałoby jeszcze podać nr adresowy danego modułu 'aa' jak w przypadku komendy „LCDaaTEST”.

Kolejne zapytanie dotyczy ustawienia wszystkich wartości na wszystkich wyświetlaczach od razu a nie tylko na pojedynczym.

7SEGMWC₀W₀...W₀, C₁W₁...W₁, C₂W₂...W₂, C₃W₃...W₃, C₄W₄...W₄, C₅W₅...W₅S<CR>

Parametry jak poprzednio oznaczają:

- c_x – wartość 0 dla świecenia ciągłego, wartość 1 dla migania, gdzie x oznacza nr cyfry
- w_x ... w_x – liczba wyświetlana na wyświetlaczu x

Zwraca standardowo „OK”.

Przykład:

7SEGMW012.3,05,111.2,00,1-5,1-99t

Przesłanie takiego zapytania spowoduje wyświetlenie na kolejnych wyświetlaczach wartości:

Nr liczby	Wartość	Miganie
0	12.3	NIE
1	5.0	NIE
2	11.2	TAK
3	0.0	NIE
4	-5.0	TAK
5	-99	TAK

Podobnie jak w poprzednim przypadku w wersji specjalnej z możliwością wybrania adresu modułu mielibyśmy komendę wymagającą dodatkowo jego podania:

7SEGMaaWC₀W₀...W₀, C₁W₁...W₁, C₂W₂...W₂, C₃W₃...W₃, C₄W₄...W₄, C₅W₅...W₅S<CR>

Następne zapytanie służy do odczytu wszystkich wartości z przetworników A/C, do których podpięte są potencjometry:

A D C a a s<CR>

Komenda ta nie wymaga specjalnego tłumaczenia. Jako „aa” podajemy adres modułu. W standardowej wersji to „00”. Zwracana wartość to:

OKW₀W₀W₀W₀W₁W₁W₁W₁W₂W₂W₂W₂W₃W₃W₃W₃W₄W₄W₄W₄W₅W₅W₅S<CR>

Tym razem w zwracanym wyniku znajduje się suma kontrolna, dzięki czemu odbierając dane możemy sprawdzić czy nie występują błędy. Jak widać mamy po 4 znaki opisujące każdą wartość widzianą przez przetworniki. Wartość ta może być od 0000 dla minimum do 1023 dla maksimum. Przykładowa zwrócona wartość:

OK001201230512102309000050q

a więc potencjometry są ustawione na:

Nr potencjometru	Wartość	Procentowo przekręcenie
0	12	1,17%
1	123	12,02%
2	512	50,04%
3	1023	100,00%
4	900	87,97%
5	50	4,89%

Przeliczenie wartości na procenty jest dość proste i polega na obliczeniu:

$$PROCENTOWO = \frac{WARTOŚĆ}{1023} * 100 \%$$

Ostatnie komendy są dla wyświetlacza LCD. Jest ich dość sporo, z różnym przeznaczeniem, więc pozostawiliśmy je sobie na koniec jak wisienka na torcie. Wyświetlacz jest połączony z przyciskami, a więc dostęp do stanu przycisków jest poprzez komendy wyświetlacza. I od tego zaczniemy:

L C D a a P R Z Y C s<CR>

Zapytanie jak zwykle zawiera adres „aa” oraz sumę kontrolną „s”. Jego zadaniem jest zwrócenie stanu przycisków. Format zwrotu jest następujący:

OKI_{środek}I_{góra}I_{dół}I_{lewo}I_{prawo}T_{środek}T_{góra}T_{dół}T_{lewo}T_{prawo}S<CR>

Tam gdzie jest wartość:

- I – to należy rozumieć, że jest to jeden impuls jeśli 1 lub 0 jeśli impulsu nie było. Te wartości dają możliwość bez wdawania się w szczegóły rozróżnianie dokładnie jednego wciśnięcia klawisza. Szczegóły przy następnej instrukcji w przykładzie
- T – trzymanie przycisku, gdzie 0 oznacza puszczone, 1-9 to czas przytrzymania, a więc jak jest 2 to przycisk jest trzymany 2-1 sekund. Ta opcja pozwala określić szybkie przewijanie jeśli przyciski mają służyć do nastawiania jakiejś wartości

Przykład. Układ zwrócił:

OK1001050048>

Z powyższego wynika, że przycisk środkowy oraz w lewo został naciśnięty impulsowo i nie został jeszcze skasowany (patrz kasowanie w następnej komendzie). Dodatkowo dowiadujemy się, że przycisk środkowy jest już trzymany od 4 sekund (bo 5-1=4), przycisk w lewo od 3 sekund (4-1=3) oraz, że przycisk w prawo jest trzymany od 7 sekund (8-1=7) ale nie ma dla „I” impulsu, więc wcześniej został skasowany i nadal jest trzymany.

Kolejna komenda służy do kasowania stanu impulsowego wszystkich przycisków (zaraz omówimy w przykładzie po co):

L C D a a P R Z Y C 0 s<CR>

Zwracane jest OK.

Komenda kasowania stanu jest niezbędna do korzystania z impulsowego obsługiwanie klawiszy. Jak to działa rozpatrzmy na prostym algorytmie, gdzie chcemy aby każde wciśnięcie przycisku do góry było rozpoznawalne jeden raz, a więc jak ktoś dłużej trzyma to i tak widoczne jest tylko jako 1 raz. Aby to zrobić postępujemy w następujący sposób:

1. Wysyłamy **LCDa a P R Z Y C s**
2. Sprawdzamy wartość I_{góra} czyli tego przycisku który nas interesuje. Jeśli jest 0 to przechodzimy do 1.
3. Jeśli jest 1 to wysyłamy **LCDa a P R Z Y C 0 s** oraz wykonujemy to co chcemy dla wciśniętego przycisku
4. Przechodzimy do 1. powtarzając cykl

Z powyższego algorytmu wynika, że chcąc rozpoznawać pojedyncze wciśnięcia najpierw czekamy aż klawisz zostanie wciśnięty, potem kasujemy jego stan i wykonujemy przypisaną dla niego operację, a WSKP PANEL już zagwarantuje, że dany klawisz nie otrzyma impulsu ponownie dopóki nie zostanie puszczone i ponownie wciśnięty.

W przypadku kiedy nie interesują nas pojedyncze przyciśnięcia a chcemy mieć bezpośrednio informację o tym czy klawisz jest wciśnięty czy nie jest i ile czasu to korzystamy z informacji „T” w zwracanej wartości.

Posługiwanie się przyciskami na panelu nie jest skomplikowane dla programisty jak widać. Przechodzimy do komend samego wyświetlacza LCD. Na początek komenda określająca ile znaków jest w kolumnie wyświetlacza LCD. Jest to komenda wynikająca z kompatybilności z innymi modułami opartymi na tym samym interfejsie.

L C D a a K O L k s<CR>

Parametrami jest znany nam adres „aa” oraz „k”, które może przyjmować wartości:

- 0 – 8 znaków w linii
- 1 – 16 znaków w linii
- 2 – 20 znaków w linii

Zwracane jest „OK”.

Z powyższego wynika, że jeśli na panelu jest wyświetlacz 16x2 znaki to wysyłamy:

LCD00KOL1B

Określenie liczby znaków w linii jest ważne, bo wyświetlacze oparte na sterowniku HD44780 mają specyficzne zarządzanie numerami znaków, więc komenda ta ułatwia posługiwanie się modułem LCD.

Sterownik obsługuje wyświetlacz LCD pozwalając wyświetlać dwa rodzaje pól:

- pola tekstowe
- napisy przesuwane

Pól tekstowych może być maksymalnie 4 o długości do 80 znaków. Napisów przesuwanych może być maksymalnie 4 o długości do 150 znaków. Pola tekstowe i napisy przesuwane ułatwiają wyświetlanie komunikatów na wyświetlaczu. Automatycznie czyszczą wyświetlacz tam gdzie jest to konieczne, automatycznie przesuwają napis tam gdzie chcemy, ogólnie

jest to bardzo fajne narzędzie.

Zajmijmy się najpierw polami tekstowymi. Zapytanie sterujące polem tekstowym wygląda następująco:

L C D a a T E K S T n f d d d p p t t ... t s <CR>

Jako parametry podajemy:

- aa – adres modułu co już znamy z wcześniejszych komend
- n – numer pola tekstowego czyli cyfra od 0 do 3
- f – aktywuj pole tekstowe jeśli 1 lub nieaktywne jeśli 0
- ddd – trzy znaki określające długość pola tekstowego (przy czym nie musi to być liczba znaków tekstu, wartość ta informuje o tym ile sterownik zarezerwuje znaków na wyświetlaczu, a więc jak tekst będzie krótszy niż długość, to pozostałe znaki zostaną uzupełnione spacjami, a jak będzie dłuższy to zostanie przycięty)
- pp – dwa znaki określające pozycję początkową pola tekstowego
- tt...tt – treść pola tekstowego

Zwracana jest wartość OK

Przykład:

LCD00TEKST0101616WKS KP PANEL7

Odczytujemy z tego zapytania, że dotyczy pola tekstowego nr 0, które zostanie włączone. Długość pola tekstowego to 16 znaków, a pole zaczynać się będzie od znaku 16 (w naszym przypadku znaki 0-15 to znaki w pierwszej linii 16-31 to znaki w drugiej linii). A więc rezerwujemy 16 znaków w drugiej linii, czyli całą drugą linię, w której wyświetli się treść

WKS KP PANEL

Jeśli policzymy znaki to widzimy, że napis ma 11 znaków, a zarezerwowaliśmy 16. Co to daje? Spójrzmy na inny przykład: Wywołujemy w odstępie 3 sekund dwie komendy:

LCD00TEKST0101616Witam y państwa!G

LCD00TEKST0101616WKS KP PANEL7

Obie komendy zaczynają się tak samo, dotyczą tego samego pola tekstowego, które jest w tym samym miejscu i ma taką samą długość. Jak zadziała układ? Najpierw wyświetli:

Witam y państwa!

A po trzech sekundach po drugiej komendzie wyświetli:

WKS KP PANEL

Oba napisy różnią się długościami, a pole tekstowe spowoduje, że cały zarezerwowany obszar będzie odświeżony wraz z nowym napisem. Niektórym osobom może się to wydawać bardzo logiczne (bo takie jest) jednak oryginalnie posługując się wyświetlaczem nie mamy dostępu do pól tekstowych a więc czyszczenie nadmiarowych znaków było zostawione programiście. Bez tego uzyskałby efekt następujący:

Witam y państwa!

I trzy sekundy później

WKS KP PANELtwa!

Pola tekstowe ułatwiają wyświetlanie tekstów i przechodzenie automatycznie do następnej linii jeśli tekst nie mieści się w jednej np. zapytanie:

LCD00TEKST0103216Witam y państwa serdecznie!h

spowoduje wyświetlenie w dwóch liniach następującej treści:

**Witam y państwa s
erdecznie!**

W oryginalnym sterowniku operacja wyświetlenia jest o niebo trudniejsza, bo druga linia nie zaczyna się zaraz po pierwszej i musielibyśmy najpierw przesłać adres pierwszej linii, potem jej treść, potem adres drugiej i treść drugiej. A więc sami musielibyśmy podzielić tekst. Tutaj tego nie musimy robić. Nie zmienia to jednak faktu, że podział wyrazów i tak pozostaje już na głowie programisty, jednak w tym przypadku to tylko kwestia dodania jednej spacji pomiędzy „państwa” i „serdecznie”.

Musimy jeszcze sobie zaznaczyć, że jeśli dwa pola tekstowe będą na siebie nachodzić tak jak w tym przykładzie wywołania dwóch komend kolejno po sobie:

LCD00TEKST0101600Witam y państwa!@

LCD00TEKST1101608WKS KP PANEL9

spowoduje problem. Oczywiście sterownik będzie działał, ale na wyświetlaczu nie otrzymamy niczego sensownego, ponieważ pole tekstowe nr 0 rezerwuje 16 znaków od pozycji 0, a więc znaki od 0 do 15 są dla pola tekstowego 0. Pole nr 1 natomiast rezerwuje 16 znaków od pozycji 8, a więc rezerwuje znaki od 8 do 23. Oba pola tekstowe jak są aktywne to nachodzą na siebie, a więc ich poprawne wyświetlanie nie jest możliwe. Jedno z pól musi być wyłączone.

Włączanie i wyłączanie pola tekstowego może odbywać się przy pomocy powyższej komendy, a można posłużyć się wersją skróconą:

L C D a a T E K S T n f s <CR>

Opis parametrów znany z wcześniejszej komendy. Jak widać w tym przypadku określamy tylko nr pola tekstowego oraz czy ma być ono włączone czy wyłączone.

Zwracane jest OK.

Następna komenda wyświetla napisy przesuwane (oczywiście po skończeniu napisu odtwarza się on od początku):

L C D a a S C R n f d d d p p r k t t ... t s <CR>

Jako parametry podajemy:

- aa – adres modułu co już znamy z wcześniejszych komend
- n – numer pola przesuwanego czyli cyfra od 0 do 3
- f – aktywuj pole przesuwane jeśli 1 lub nieaktywne jeśli 0
- ddd – trzy znaki określające długość pola przesuwanego (przy czym nie musi to być liczba znaków tekstu, wartość ta informuje o tym ile sterownik zarezerwuje znaków na wyświetlaczu, a więc tekst może być dłuższy i będzie przesuwany automatycznie)
- pp – dwa znaki określające pozycję początkową pola tekstowego
- r – podanie 1 oznacza, że napis ma być przesuwany od początku, 0 jeśli ma być kontynuowane wyświetlanie od tej pozycji co była obecnie (przypadek kiedy chcemy zmieniać treść napisu przesuwanego bez konieczności jego ciągłego odtwarzania od początku np. wyświetlanie daty i godziny jako napis przesuwany ma sens, ale pod warunkiem, że napis będzie szedł cały czas a my modyfikujemy jedynie treść)
- k – tempo przesuwania od 0 do 9 co odpowiada tempu $0,07\text{sek} * (k+1)$, a więc podając 3 uzyskamy przesuwanie znaków co $0,07*4\text{sek} = 0,28\text{sek}$ będzie przesuwany się znak.
- tt...tt – treść pola tekstowego

Zwracana jest wartość OK

Przykład:

LCD00SCR010160013Witamy Państwa w panelu WSKSP PANEL! Pozdrowienia od ELKOSMITO!>

W tym przypadku uzyskamy napis, który będzie się przesuwał w całej pierwszej linii (bo zarezerwowaliśmy 16 znaków począwszy od znaku 0). Znaki będą przesuwały się co 0,28sek, a napis po uruchomieniu komendy zacznie być przewijany od początku. Nie musimy podawać znaków spacji na końcu ani na początku, bo układ sam doda kilka spacji, żeby napis był czytelniejszy a jego początek od końca się odróżniał.

Tak jak w przypadku pola tekstowego tak i tutaj trzeba pamiętać aby napisy nie zachodziły na siebie, bo to bez sensu. Włączanie i wyłączanie napisów przesuwanych odbywa się przy pomocy powyższej komendy lub skróconej:

L C D a a S C R n f s <CR>

Zwracana wartość to OK, a parametry wiemy z poprzedniej komendy.

Teraz chwila relaksu z komendą czyszczenia ekranu, która dodatkowo wyłącza wszystkie pola tekstowe i napisy przesuwane. Zapytanie to wygląda następująco:

L C D a a C Z Y S C s <CR>

Zwracana wartość to OK.

Jeśli chcemy wykorzystać polskie znaki w tekście to możemy wykorzystać komendę uruchamiającą 8 dużych znaków polskich (wyświetlacze HD44780 pozwalają wyświetlać maksymalnie 8 znaków użytkownika w danej chwili):

L C D a a P L s <CR>

Zwracana wartość to OK.

Wywołanie tej komendy powoduje, że możliwe będzie posługiwanie się w tekście dużymi znakami z polskiego alfabetu: „ĄĆĘŁŃÓŚŻ”. Aby znaki były wyświetlane poprawnie to prócz komendy musimy zadbać aby kodowanie przesyłanego zapytania było zgodne z „Windows-1250” (szczegóły można znaleźć w internecie na temat tego czym jest strona kodowa).

Chcąc skorzystać z małych polskich liter możemy wywołać komendę:

L C D a a p l s <CR>

Zwracana wartość to OK.

Po tej komendzie mamy dostęp do znaków „ąćęłńóśż”

Jeśli w jednej chwili zachodzi potrzeba wykorzystania znaków zarówno polskich małych jak i dużych, to musimy określić maksymalnie 8 znaków, które mogą być jednocześnie wyświetlane na wyświetlaczu oraz wywołać komendę:

LCDaaZESTAWz₁z₂z₃z₄z₅z₆z₇z₈S<CR>

Zwracana wartość to OK.

Jako znaki z1-z8 możemy podać następujące symbole: „, ą ć ę ł ń ó ś ź ż Ą ć ę ł ń ó ś ź ż”. Można podać maksymalnie 8 znaków (może być mniej), którymi potem będziemy posługiwać się podczas wyświetlania tekstów. Przy tej okazji możemy poprzednie próby z polami tekstowymi oraz napisami przesuwanymi udoskonalić, żeby polskie znaki wyświetlały się w tych próbach poprawnie.

Czasami jednak zachodzi potrzeba wykorzystania innego znaku spoza liter. Moduł zapewnia dostęp do kilku znaków dodatkowych. Do ich wykorzystania podobnie jak znaków polskich służy zapytanie:

LCDaaZNAKIuupzs<CR>

Zwracana wartość to OK.

Parametry:

- aa – adres modułu
- uu – dwa znaki numeru symbolu z tabeli poniżej
- p – pozycja w pamięci wyświetlacza, pod którą znajdzie się wybrany symbol (od 0 do 7)
- z – znak pod którym symbol będzie dostępny (patrz przykłady)

Adres modułu nie wymaga tłumaczenia. Zapoznajmy się symbolami jakie są dostępne i wbudowane w sterownik WSKSP PANEL:

<i>Nr symbolu</i>	<i>Znak</i>	<i>Uwagi</i>
0	ą	
1	ć	
2	ę	
3	ł	
4	ń	
5	ó	
6	ś	
7	ź	
8	Ą	
9	Ć	
10	Ę	
11	Ł	
12	Ń	
13	Ó	
14	Ś	
15	Ź	
16	°	Znak stopni
17	→	Strzałka w prawo
18	←	Strzałka w lewo
19	↑	Strzałka w górę
20	↓	Strzałka w dół
21	°C	Stopnie Celcujusza w jednym znaku (trochę obogo, ale przydatne czasami)
22	√	Tzw. ptaszek (trochę ubogi, ale przydatny czasami)
23	Ż	
24	ż	

Przykład:

LCD00ZNAKI164^>

Przykład pobiera znak o nr 16 (czyli znaczek stopni) i umieszcza je w pamięci wyświetlacza na pozycji 4 (pozycja jest ważna, bo umieszczając kilka znaków musimy określić co będziemy zastępować). Dodatkowo znak będzie dostępny podczas pisania pod symbolem „^”. Przykład pola tekstowego:

LCD00TEKST0101600Temp: 10^CI

Takie pole tekstowe wyświetli w pierwszej linii wyświetlacza treść:

Temp: 10°C

Jeśli chcemy sami złożyć znak, którego nie ma, to możemy go przesłać do wyświetlacza przy pomocy komendy:

LCDaaSYMBpzb₀b₀b₁b₁b₂b₂b₃b₃b₄b₄b₅b₅b₆b₆b₇b₇**s<CR>**

Zwraca OK.

Parametry:

- aa – adres modułu
- p – pozycja w pamięci wyświetlacza, pod którą znajdzie się wybrany symbol (od 0 do 7)
- z – znak pod którym symbol będzie dostępny (patrz wcześniejszy przykład)
- b₀b₀– b₇b₇ – kolejne bajty zapisane w systemie szesnastkowym (litery duże) opisujące znak

Komenda jest komendą dla bardziej zaawansowanych użytkowników, którym ona nie sprawi większych problemów w zrozumieniu jak to działa. Nie będziemy jej w związku z tym analizować szczegółowo. Pokażemy sobie jedynie przykład strzałki (takiej samej jak w tabeli, ale tym razem utworzonej przy pomocy tej komendy):

LCD00SYMB2>0004021F02040000^

Strzałka zostanie zapisana w pamięci wyświetlacza pod nr 2, podczas pisania będzie dostępna pod symbolem „>”.

Ostatnim elementem, który bywa niezmiernie przydatny jest kursor. Aby pojawił się migający kursor należy użyć zapytania:

L C D a a K U R p p s <CR>

Tradycyjnie zwracana jest wartość OK.

Parametry to:

- aa – adres modułu
- pp – pozycja znaku kursora

Wyłączenie kursora jest podobne do włączenia i należy podać:

L C D a a K U R - - s <CR>

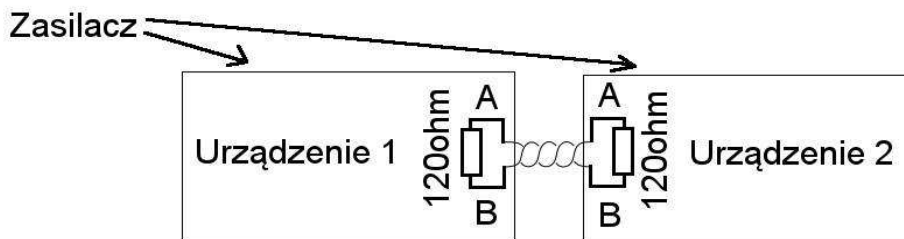
czyli dwa znaki – w miejscu pozycji.

Kursor miga jako ciemny kwadracik i nie jest przesuwany automatycznie. Programista może określić gdzie będzie znajdował się kursor na wyświetlaczu. Oczywiście umieszczenie kursora w polu tekstowym nie stanowi problemu.

I to właściwie wszystko jeśli chodzi o interfejs i protokół transmisji. Jak widać nie ma tutaj właściwie nic skomplikowanego jeśli się zna podstawy interfejsów komputera.

Dodatek A: Instalacja sieci RS485

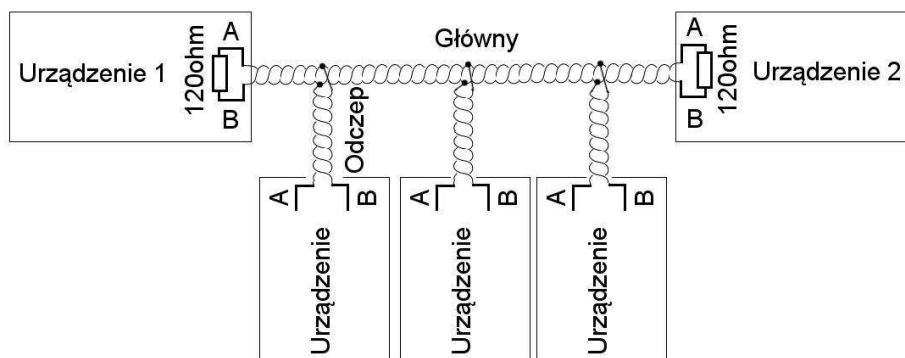
Sieć RS485 wymaga do połączenia skręconych przewodów pomiędzy urządzeniami. Przedstawia to poniższy rysunek.



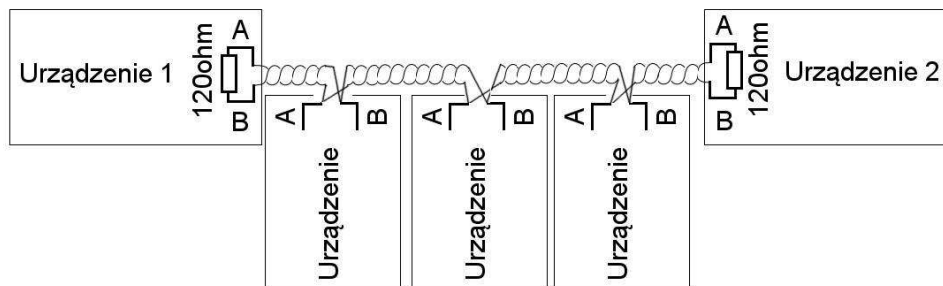
Jeśli w instalacji mamy tylko 2 urządzenia to najprościej podłączyć je w ten sposób. Połączenia powinniśmy dokonać przewodem tzw. skrętką. Może to być skrętka komputerowa. W zależności od warunków pracy i odległości skrętka może być bez ekranu lub ekranowana. Jeśli środowisko pracy będzie z dużymi zakłóceniami lub odległości między modułami są znaczne to wtedy zalecić można skrętkę ekranowaną. Na powyższym rysunku zasilacz jest wspólny dla obu urządzeń. Może się jednak zdarzyć, że urządzenia będą miały osobne zasilacze. W takim przypadku najlepiej jeszcze połączyć oba zasilacze wspólną masą, ponieważ zwłaszcza pomiędzy zasilaczami impulsowymi może występować różnica potencjałów masy i może ona powodować nieprawidłową pracę urządzeń a nawet ich uszkodzenie. Jeśli w instalacji mamy przewód uziemiający, a nie chcemy puszczać dodatkowego przewodu wspólnej masy pomiędzy zasilaczami to możemy połączyć GND obu zasilaczy i urządzeń do uziemienia.

Gdy w instalacji są tylko 2 urządzenia to na wejściach A i B w każdym urządzeniu powinien być rezystor dopasowany do przewodu. W przypadku skrętki komputerowej jest to rezystor 120 ohm. Jeśli urządzenie jest wyposażone w wewnętrzny rezystor 120 ohm i można go włączyć to wystarczy to zrobić. Jeśli natomiast nie, to koniecznie trzeba zapiąć rezystor zewnętrzny.

Jeśli w instalacji jest więcej niż 2 urządzenia istnieją trzy sposoby na ich podłączenie. Spójrzmy najpierw na poniższy rysunek:



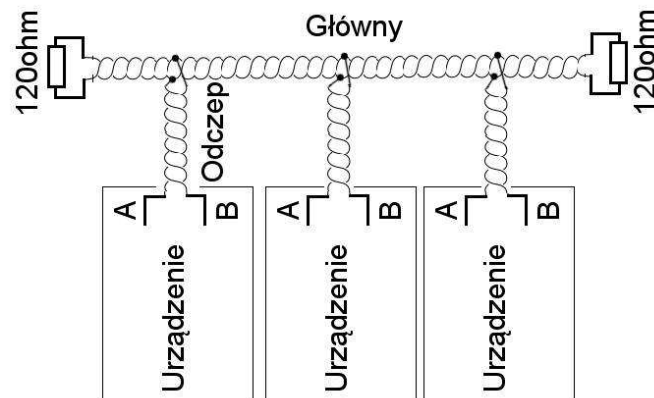
Widzimy tutaj, że mamy skrętkę główną. Główna skrętka łączy pierwsze i ostatnie urządzenie tak jak poprzednio. Na początku i na końcu jest rezystor 120ohm, czyli na urządzeniu 1 i 2 założone są opcjonalne zworki rezystorów 120 ohm lub zewnętrzne rezystory 120ohm. Główna skrętka może być długa. Do połączenia jej z kolejnymi urządzeniami wykonywane są odczepy. Odczepy powinny być wykonane również ze skrętki, jednak ich długość nie powinna być większa niż kilkadziesiąt centymetrów. Zbyt duża odległość spowodować może problemy w transmisji. Urządzenia podpięte do odczepów nie mają rezystorów 120 ohm. Oczywiście widzimy tutaj, że jeśli odczepy skrócimy do minimum to uzyskamy po prostu takie połączenie:



Tak więc jeśli jest taka możliwość to po prostu urządzenia wpinamy bezpośrednio w główną skrętkę wykonując jej rozcięcie. Nie zawsze jednak jest to możliwe i wtedy stosujemy odczepy. W większości przypadków przy instalacji w jednej szafie wszystkich urządzeń można zrezygnować z odczepów. Ważne jest aby wszystkie połączenia i odczepy były możliwe

krótkie, czyli jeśli skrętkę rozcinamy/odkręcamy to aby nie były to długie odcinki tylko jak najkrótsze.

Trzecim ostatnim sposobem połączenia urządzeń jest połączenie, w którym wszystkie urządzenia połączone są odczepami z główną skrętką, a rezystory 120ohm znajdują się na końcu głównej skrętki i są elementami zewnętrznymi. Czasami takie podłączenie jest korzystne i przedstawia je poniższy rysunek. Korzyścią z takiego połączenia jest to, że możemy rozciągnąć skrętkę po wszystkich pomieszczeniach nawet tam gdzie nie mamy zainstalowanych jeszcze modułów. Potem jak będziemy chcieli to możemy te moduły w każdej chwili dodawać.



Na koniec kilka najważniejszych uwag:

- na początku końca skrętki powinien znajdować się rezystor 120ohm (chyba, że skrętka jest niestandardowa)
- przewody A we wszystkich urządzeniach łączymy do A i przewody B łączymy do B
- jeśli urządzenia są zasilane z różnych zasilaczy to zaleca się aby masa wszystkich urządzeń była wspólna (można w tym celu wykorzystać przewód uziemiający)
- skrętka komputerowa ma 4 pary przewodów (razem 8 przewodów), a więc 6 z nich pozostanie niewykorzystane, bo tylko 2 z nich będą potrzebne do transmisji. Wolne 6 można połączyć po 3 i wykorzystać do puszczenia zasilania modułów ale w przypadku wielkich odległości może to nie wystarczyć oraz wykorzystanie wolnych przewodów skrętki do zasilania modułów może powodować zakłócenia i o tym warto pamiętać
- instalacja RS485 na niewielkich odległościach do 50 metrów nie jest trudna jeśli zachowa się podstawowe zasady połączeń

Uwagi

Uwaga! Urządzenie elektroniczne! Nieprawidłowe użytkowanie urządzenia może grozić uszkodzeniem odbiornika lub innymi poważniejszymi konsekwencjami w tym porażeniem prądem! Zachowaj szczególną ostrożność!