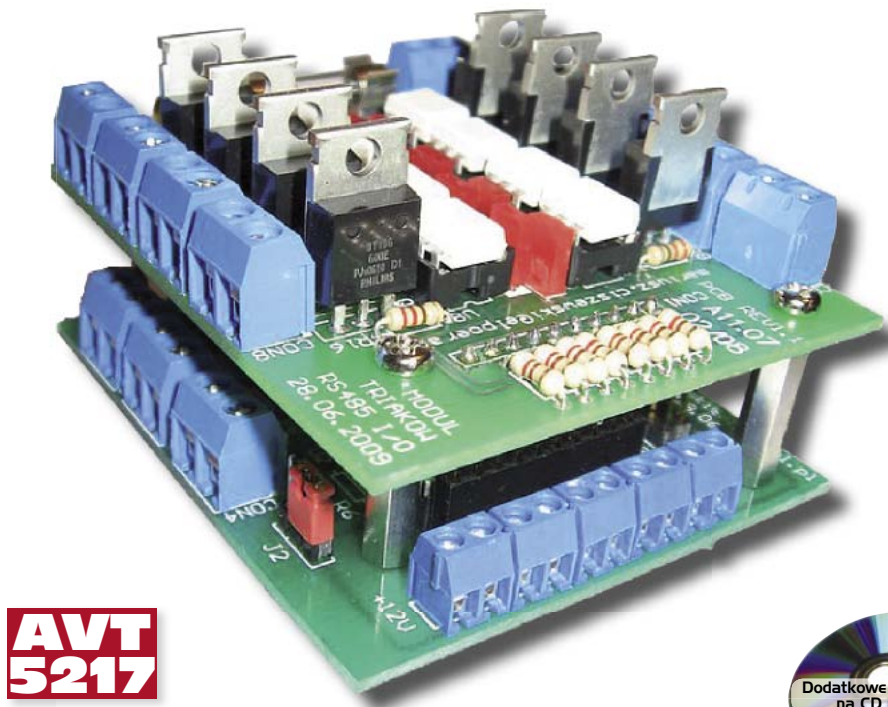


# Uniwersalna karta wyjść z interfejsem RS485 i mikrokontrolerem AVR

Uniwersalna karta wykonawcza z interfejsem RS485 przeznaczona do wykorzystania m. in. w systemach automatyki domowej, gdzie zachodzi konieczność sterowania urządzeniami na dużych odległościach. Oparta na mikrokontrolerze AVR i zawierająca złącze programowania w systemie (ISP) – może być atrakcyjną gratką nie tylko dla miłośników środowiska BASCOM. W zależności od zastosowanego modułu, elementami wykonawczymi mogą być przekaźniki lub triaki (sterowanie urządzeniami sieci elektrycznej), bądź bezpośrednio układ ULN2803, obecny na płycie sterownika, zapewniający na wyjściach poziomy napięcie 0 V lub 12 V i wydajność prądową o wartości 500 mA.

**Rekomendacje:** uniwersalny moduł mogący znaleźć zastosowania w układach automatyki budynków, przemysłowej itp.

Karta została pomyślana jako element końcowy (wykonawczy) możliwy do zastosowania w dowolnych systemach wykorzystujących przewodowe łącze RS485. Przykładem takiego systemu sterowania może być komputer PC z konwerterem RS232<->RS485 (np. AVT530) z dołączonymi do magistrali wieloma kartami wykonawczymi. Standard RS485 w swej podstawowej wersji przewiduje współpracę 32 urządzeń, na magistrali (tor dwuprzewodowy, różnicowy) o długości do 1200 m (długość optymalna dla systemów wykorzystujących w torze transmisji danych napięcie różnicowe). Maksymalna liczba 32 obsługiwanych urządzeń



**AVT  
5217**

w wielu przypadkach jest zupełnie wystarczająca. Ograniczenie liczby urządzeń pracujących na magistrali wynika ze skończonej wydajności energetycznej nadajnika oraz impedancji pojedynczego odbiornika, która w standardzie RS485 wynosi typowo 12 kΩ. Sumaryczna impedancja na linii (pomijając wpływ rezystorów terminujących) nie może wynieść mniej niż 375 Ω, stąd istnieje możliwość podłączenia do sieci maksymalnie 32 urządzeń. Istnieją jednak układy o większej impedancji odbiorników, na których można zbudować sieć z większą liczbą współpracujących urządzeń, np. MAX487 (impedancja odbiornika: 48 kΩ, możliwość podłączenia do 128 urządzeń) lub MAX3072 (impedancja odbiornika: 96 kΩ, możliwość podłączenia do 256 urządzeń).

Zwiększanie liczby urządzeń pracujących na magistrali, np. za pomocą powyższych układów, nie jest oczywiście całkiem bezkarne i odbywa się kosztem konieczności zmniejszenia prędkości transmisji danych (układy MAX487, MAX3072 pracują prawidłowo z prędkością do 250 kbps, podczas gdy standardowy MAX485 pracuje prawidłowo

#### AVT-5217 w ofercie AVT:

AVT-5217A – płytka drukowana  
AVT-5217B – płytka drukowana + elementy

#### Dodatkowe materiały na CD i FTP:

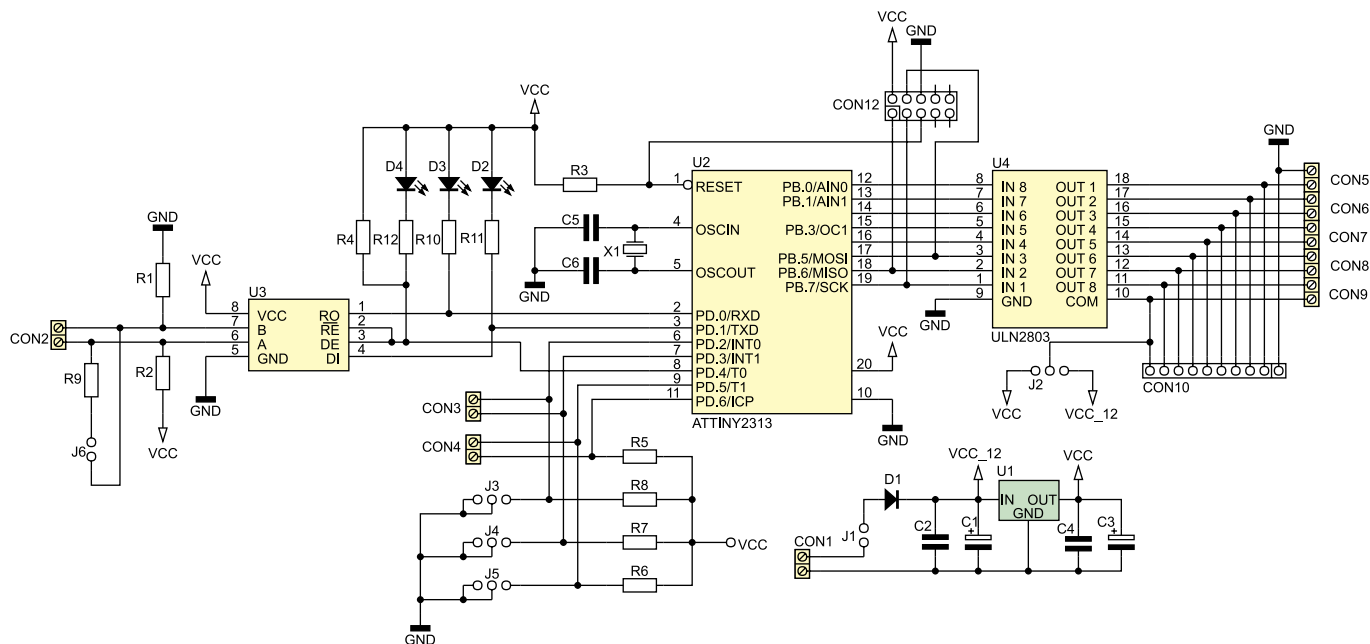
<ftp://ep.com.pl>, user: 18366, pass: 3scpp470

- wzory płytek PCB
- program
- projekty pokrewne
- karty katalogowe i noty aplikacyjne elementów oznaczonych na Wykazie Elementów kolorem czerwonym

#### Projekty pokrewne na CD i FTP:

(wymienione artykuły są w całości dostępne na CD)

- AVT-814 Konwerter RS232 – RS485 z optymalizacją (EP 5/1999)
- AVT-5006 Dwukierunkowy interfejs RS232/RS485 (EP 4/2001)
- AVT-5098 Konwerter USB – RS485 z separacją galwaniczną (EP 2/2003)
- AVT-530 Konwerter RS232 – RS485 (EP 6/2003)
- AVT-531 Karta wyjść przekaźnikowych (EP 7/2003)
- AVT-532 Karta wyjść optoizolowanych (EP 7/2003)
- AVT-533 Karta wyjść cyfrowych (aktywne GND) (EP 8/2003)
- AVT-534 Karta wyjść cyfrowych (aktywne VCC) (EP 8/2003)
- AVT-535 Karta wyjść cyfrowych (EP 9/2003)
- AVT-536 Karta wyjść analogowych (EP 9/2003)
- AVT-537 Moduł terminala z wyświetlaczem LED (EP 10/2003)



Rys. 1. Schemat ideowy układu sterownika

wo z prędkością do 2,5 Mbps). RS485 wymaga stosowania rezystorów terminujących na obu krańcach magistrali, których zadaniem jest dopasowanie impedancji linii, a tym samym tłumienie niepożądanych odbić fali na końcach toru transmisyjnego. Wartość rezystora terminującego wynosi typowo 120 Ω.

## Opis konstrukcji

Prototyp opisanego w artykule urządzenia zbudowano z użyciem popularnego układu MAX485 z typową dla standardu RS485 impedancją odbiornika równą 12 kΩ. Oznacza to, że w jednym systemie fizycznie może pracować np. 31 takich kart oraz jedno urządzenie sterujące (np. komputer PC z konwerterem RS232 <-> RS485). Urządzenie po-

siada modułową budowę: składa się z dwóch płytek PCB: płytki mikrokontrolera oraz płytki układu wykonawczego. Sterownik został zaprojektowany w taki sposób, aby zapewnić możliwość wykorzystania, w zależności od potrzeb, różnych układów wykonawczych. Na tę chwilę zaprojektowałem dwa układy wykonawcze – pierwszy na przekaźnikach, drugi na triakach. Jeżeli zaś chcemy sterować urządzeniami o napięciu zasilania równym 12 V i poborze prądu nieprzekraczającym 500 mA, możemy zrezygnować ze stosowania modułu wykonawczego, wykorzystując listwę wyjść sterujących płytki sterownika.

**Sterownik.** Schemat ideowy sterownika pokazano na **rys. 1**. Znajdują się tu: blok stabilizacji napięcia wejściowego 5 V – U1 (7805) w otoczeniu podstawowej aplika-

cji kondensatorów pomocniczych, U2 tj. mikrokontroler ATtiny2313 interpretujący dane napływające z sieci i sterujący na ich podstawie elementami wykonawczymi, jak również generujący dane do wysłania w sieć, U4 (ULN2803) pełniący rolę bufora sterującego układami wykonawczymi oraz U3 (MAX485) dopasowujący sprzętowy port szeregowy UART mikrokontrolera do wymogów elektrycznych standardu RS485. Na płytce mikrokontrolera dostępne jest również złącze CON12 pozwalające na programowanie mikrokontrolera bezpośrednio w układzie, bez wyciągania mikrokontrolera z podstawki, za pomocą powszechnie dostępnych dziś programatorów ISP (np. AVT 2550/P).

Złącze CON10 służy do podłączenia modułów wykonawczych z przekaźnikami lub triakami. Złącza CON3 i CON4 są dodatkowymi wejściami dowolnego przeznaczenia (o ile zrezygnujemy z mechanicznego ustawiania adresu urządzenia w sieci za pomocą zworek J3, J4, J5). Złącza CON5, CON6, CON7, CON8, CON9 służą do bezpośredniego podłączenia obwodów sterowanych przez

Na CD: karty katalogowe i noty aplikacyjne elementów oznaczonych na wykazie elementów kolorem czerwonym

### Wykaz elementów Płytki sterownika

#### Rezystory:

R1, R2: 510 Ω  
R3...R8: 4,7 kΩ  
R9: 120 Ω  
R10...R12: 150 Ω

#### Kondensatory:

C1: 470 μF/16 V  
C2, C4: 100 nF  
C3: 47 μF/6,3 V  
C5, C6: 33 pF

#### Półprzewodniki:

U1: 7805  
U2: ATtiny2313  
U3: MAX485  
U4: ULN2803  
D1: 1N4001  
D2: LED 5 mm zielona  
D3: LED 5 mm czerwona  
D4: LED 5 mm żółta

#### Inne:

X1: kwarc 8 MHz  
CON1...CON4: ARK2 5 mm  
CON5...CON9: ARK2 3 mm  
CON10: gniazdo na goldpiny 10×1

CON12: goldpin 5×2

J1: goldpin 2×1

J2...J5: goldpin 3×1

ZW1...ZW7: zworki (druć kynarowy)

#### Płytki przekaźników

#### Półprzewodniki:

D1...D6: 1N4001

#### Inne:

PK1...PK6: przekaźniki JZC12F  
CON1: goldpin 10×1 wysokie: h = 20 mm  
CON2...CON7: ARK3 5 mm

#### Płytki triaków

#### Rezystory:

R1...R16: 220 Ω  
R17...R24: 820 Ω  
R25: 100 kΩ

#### Półprzewodniki:

U1...U8: MOC3021 (optotriak)  
Q1...Q8: BT136 (triak)  
D1...D8: dioda LED czerwona, prostokątna  
D9: 1N4001  
D10: dioda LED 5 mm żółta

#### Inne:

CON1...CON9: ARK2 5 mm  
CON10: goldpin 10×1 wysokie: h = 20 mm  
F1: Bezpiecznik 4A + oprawki do druku

R E K L A M A

# STM32 FanClub

32-bitowe mikrokontrolery z rdzeniem Cortex-M3  
Krzysztof Paprocki

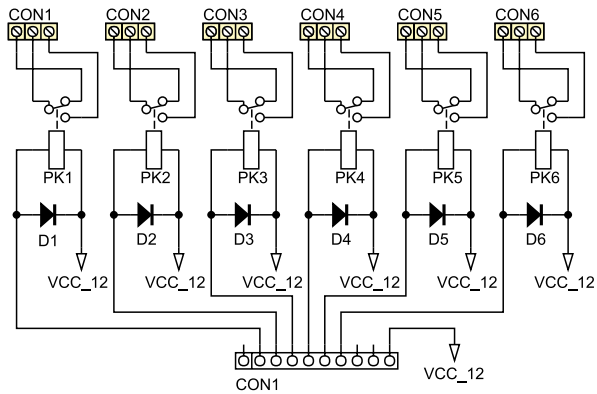
## Mikrokontrolery STM32 w praktyce

**Pierwsza na świecie książka o STM32**

Dostępna m.in. w **KAMAMI**  
www.kamami.pl




# STM32



Rys. 2. Schemat ideowy układu wykonawczego z przełącznikami

kartę (bez udziału zewnętrznego modułu układów wykonawczych), złącze CON2 do podłączenia magistrali RS485 oraz złącze zasilania CON1. Diody D1 zabezpiecza układ przed niepoprawnym podłączeniem biegunów zasilania, natomiast jumper J1 pozwala na łatwe odłączenie (bez fizycznego odłączenia kabli) z sieci opartej na RS485 urządzenia, które np. przestało działać poprawnie i w wyniku tego np. zakłóca prawidłową transmisję danych poprzez magistralę.

nych w sieć. W ten sposób wyeliminowano problem zabronionego nadawania przez więcej niż jedno urządzenie w jednej chwili czasu, które zablokowałoby prawidłową transmisję danych.

Jumper J6 pozwala na dołączenie lub odłączenie rezystora terminującego na jednej lub po obu stronach magistrali (w fizycznie pierwszym i/lub ostatnim urządzeniu podpiętym na całą długość linii). Jego dołączenie może mieć znaczenie w przypadku za-

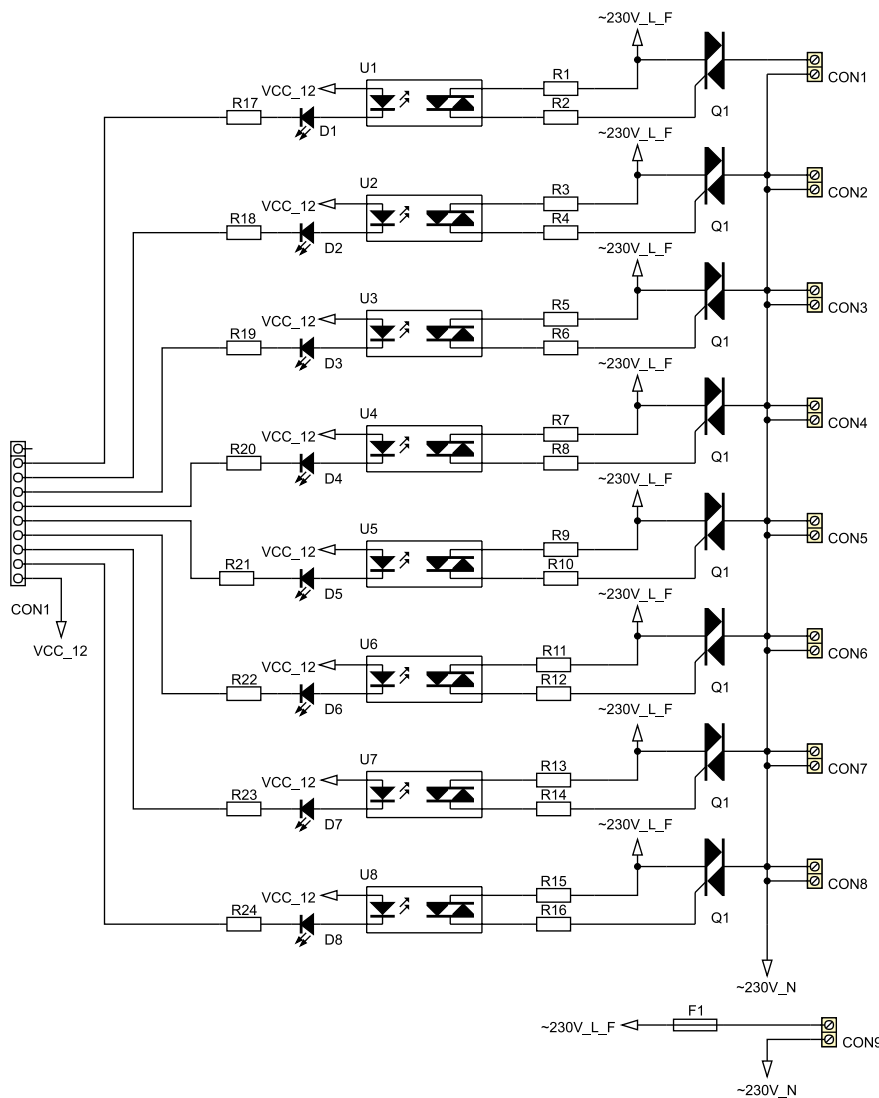
Jumpery J3, J4 oraz J5 umożliwiają podłączenie do masy wyprowadzeń mikrokontrolera, odpowiednio PD2, PD3 i PD5, domyślnie podciągniętych (za pomocą rezystorów R6...R8) do plusa zasilania. Program zaszyty w mikrokontrolerze, po podłączeniu zasilania odczytuje te ustawienia i interpretuje jako adres własny danego urządzenia w sieci. Na podstawie tego adresu program wyznacza również czas opóźnienia przed wysłaniem danych

obserwowania błędów w transmisji danych, zwłaszcza przy dużej prędkości transmisji danych oraz występowaniu wielu urządzeń w sieci. Z reguły jednak jego stosowanie nie jest konieczne, czasem wręcz zaburza poprawną pracę sieci. Na płytce sterownika znajdują się również diody LED: D2, D3 oraz D4, pozwalające na bieżące kontrolowanie stanu magistrali (odpowiednio DI – Driver Input, do której podłączona jest linia TXD mikrokontrolera, RO – Receiver Output, do której podłączona jest linia RXD mikrokontrolera) dioda D4 sygnalizuje natomiast stan linii sterującej układu MAX485 (sygnały ~RE – Receiver Output Enable aktywny w stanie niskim oraz DE – Driver Output Enable aktywny w stanie wysokim). Ponieważ stany aktywne obu tych sygnałów mają przeciwną wartość logiczną można było połączyć je i sterować za pomocą jednej linii. Gdy D4 świeci MAX485 znajduje się w stanie odbioru, a gdy dioda D4 pozostaje wygaszona – nadawania. Rezystor R3 odpowiedzialny jest za wygenerowanie sygnału RESET po podłączeniu zasilania, a jumper J2 pozwala na wybranie odpowiedniego napięcia, jakie ma pojawiać się na wyjściach układu sterownika (dla obu przygotowanych modułów wykonawczych – na triakach oraz na przełącznikach prawidłową wartością będzie 12 V, czemu odpowiada ustawienie jumpera J2 w pozycji 2-3.

**Moduł wykonawczy z przełącznikami.**

Moduł wykonawczy z przełącznikami (rys. 2) zawiera niewiele elementów: 6 popularnych, łatwych do kupienia i tanich przełączników JZC-20F, 6 diod wpinanych równolegle z cewką przełącznika, eliminujących przepięcia generujące się w cewce oraz 6 potrójnych złączy śrubowych typu ARK umożliwiających dowolne wykorzystanie styku przelotowego przełącznika. Moduł obsługuje tylko 6 pierwszych kanałów. Pozostałe dwa można wykorzystać z poziomu złącza wyjść sterownika, do sterowania urządzeniami zasilanymi napięciem 12 V. Tyle przełączników udało się zmieścić na płytce gabarytowo odpowiadającej rozmiarom płytki sterownika. Nic nie stoi jednak na przeszkodzie by zaprojektować PCB dla modułów z innymi np. mniejszymi przełącznikami.

**Moduł wykonawczy z triakami.** Moduł triaków (rys. 3) jest nieco bardziej skomplikowany, niż moduł z przełącznikami. Ma osiem torów LED → optotriak → triak oraz bezpiecznik, złącze śrubowe CON9 służące do podłączenia napięcia sieciowego 230 VAC oraz 8 wyjść CON1...CON8, do których można podłączyć sterowane urządzenia (np. żarówki, wentylatory i inne, pracujące przy zasilaniu napięciem przemiennym 230 VAC). Ze względu na przeznaczenie (sterowanie urządzeniami domowymi o niewielkim poborze prądu), zrezygnowałem ze stosowania radiatorów na triakach, jednak w przypadku



Rys. 3. Schemat ideowy układu wykonawczego z triakami

konieczności sterowania większymi obciążeniami, te radiatory trzeba będzie założyć. W celu włączania kanału, mikrokontroler wystawia stan wysoki na wejściu układu ULN2803. Na jego wyjściu pojawia się stan niski względem 12 V i trafia na wejście modułu wykonawczego. Tam przez rezystor (R7...R24) trafia do katody szeregowo podłączonej diody LED (D1...D8) załączając jej świecenie i pozwalając w ten łatwy sposób stwierdzić, czy w danym momencie kanał jest włączony – i czy zaświecona jest odpowiednia dioda optotriaka (U1...U8).

Optotriak pełni rolę separacji galwanicznej, oddzielając od niskonapięciowej części układu niebezpieczne napięcie sieci 230 VAC. Za pośrednictwem optotriaka oraz rezystorów (R1...R16) napięcie sieciowe jest podawane na bramkę triaka (Q1...Q8), a ten włącza urządzenia podłączone do złącz CON1...CON8.

Ze względu na ograniczenia gabarytowe zastosowano tylko jeden, wspólny bezpiecznik F1. Nic jednak nie stoi na przeszkodzie aby w obwodach podłączanych do złącz zastosować oddzielne bezpieczniki.

Diody LED D9 zasilana z sieci 230 VAC za pośrednictwem diody D10 oraz rezystora R25 sygnalizuje obecność napięcia sieci na złącz CON9.

## Opis programu sterującego

Po włączeniu zasilania i automatycznym wygenerowaniu się sygnału RESET, mikrokontroler odczytuje stan jumperów J3, J4 i J5. Na podstawie ich ustawienia ustala zmiennej *Adres* nadawana jest odpowiednia wartość z zakresu 0...7. Oznacza to, że za pomocą zworek jesteśmy w stanie w sposób jednoznaczny zaadresować nie więcej niż 8 kart wykonawczych (8 kart po 8 urządzeń daje możliwość sterowania 64 urządzeniami). Jeśli chcemy sterować większą liczbą kart wykonawczych pełniej wykorzystując możliwości magistrali RS485 (32 urządzenia) możemy, modyfikując nieco kod źródłowy programu napisanego w Bascom AVR, zrezygnować z ustalania adresu na podstawie zworek i przypisać go zmiennej statycznie.

Gdy adres urządzenia zostanie ustalony, mikrokontroler wyśle w sieć komunikat zgłoszeniowy w postaci tekstu: Urz: X : Gotowy > [karta][urządzenie][stan] (X oznacza numer urządzenia (zmienna *Adres*), natomiast tekst [karta][urządzenie][stan] jest zachętą do wprowadzenia kodu sterującego. Zanim jednak mikrokontroler wyśle w sieć tekst zgłoszenia, odczeka on czas ustalony na podstawie swojego adresu. Dzięki temu wyeliminowana została nieunikniona kolizja, w przypadku gdyby po dostarczeniu zasilania wszystkie urządzenia zaczęły nadawać w jednym czasie. Ponieważ każde z urządzeń wyśle komuni-

kat po odczekaniu właściwego dla swego adresu czasu, urządzenia będą zgłaszały się po kolei i kolizja nie nastąpi. Odpowiada za to fragment kodu:

```
For Opoznienie$ = 1 To Adres
    Waitms 150
Next Opoznienie$
```

Jak widać urządzenia będą się zgłaszały kolejno, w 150 ms odstępach czasu. Po wysłaniu opisanego wyżej zgłoszenia i tekstu zachęty mikrokontroler przechodzi w tryb oczekiwania na napływające dane. Wymagany jest kod sterujący składający się z trzech cyfr. Pierwsza to numer docelowej karty wykonawczej (1...8 oraz cyfra specjalna 9 oznaczająca „wszystkie”), druga cyfra oznacza numer urządzenia (1...8 oraz cyfra specjalna 9 również i w tym przypadku oznaczająca „wszystkie”), trzecia zaś cyfra może przyjąć tylko dwie wartości: 1 oznaczająca „włącz” oraz 0 oznaczająca „wyłącz”.

Nietrudno domyślić się, że wysłanie przez sieć rozkazu „131” powoduje włączenie przekaźnika numer 3 na pierwszej karcie, rozkazu „291” spowoduje włączenie wszystkich przekaźników na karcie numer 2, zaś rozkaz „990” wyłączy wszystkie przekaźniki na wszystkich kartach. Rozwiązanie takie zapewnia komfortowe sterowanie pojedynczymi urządzeniami, całymimi kartami jak i wszystkimi urządzeniami w sieci jednocześnie, za każdym razem za pomocą jednego trzy cyfrowego rozkazu sterującego.

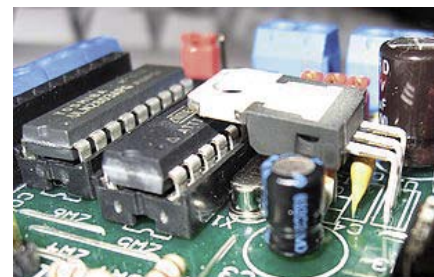
Po odebraniu rozkazu sterującego następuje tzw. parsowanie odebranego ciągu znaków i przypisanie zmiennym odpowiednich wartości. Odpowiada za to poniższy fragment kodu:

```
Input Rozkaz
    Zmienna_interpretacyjna =
    Left(rozkaz , 1 )
    Numer_karty$ = Val(zmienna_
    interpretacyjna)
    Zmienna_interpretacyjna =
    Mid(rozkaz , 2 , 1)
    Numer_urzadzenia$ =
    Val(zmienna_interpretacyjna)
    Zmienna_interpretacyjna =
    Mid(rozkaz , 3 , 1)
    Stan$ = Val(zmienna_interpre-
    tacyjna)
```

Na podstawie ustalonych w powyższy sposób zmiennych dokonywane są proste operacje logiczne:

Jeśli numer karty odpowiada naszemu adresowi albo ma wartość 9 to:

- jeśli stan jest równy 1 to:
  - Włącz urządzenie <numer urządzenia>.
  - Jeśli <numer urządzenia> jest równy 9, to włącz wszystkie urządzenia).
- Jeśli stan jest równy 0 to:
  - Wyłącz urządzenie <numer urządzenia>.



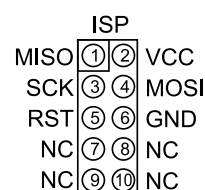
Fot. 4. Montaż stabilizatora U1

- Jeśli <numer urządzenia> jest równy 9, to wyłącz wszystkie urządzenia)

Po wykonaniu tych operacji program wraca do początku (czeka na kolejne wprowadzane rozkazy i ewentualnie je wykonuje, jeśli są adresowane do niego, bądź ignoruje, jeśli kierowane są do innych urządzeń) i wykonuje się w nieskończonej pętli.

## Montaż i uruchomienie

**Sterownik.** Płytkę sterownika została wykonana jako jednostronna, co spowodowało konieczność wykonania 7 zwór, od których należy rozpocząć montaż. Są one zaznaczone na płytce oznaczeniami ZW1...ZW7. Najwygodniej wykonać je z drutu kyanarowego. W drugiej kolejności warto zamontować wszystkie rezystory oraz diodę D1. Dalej: rezonator kwarcowy (jeśli jest niskoprofilowy), podstawki pod układy scalone, kondensatory ceramiczne, diody LED, złącza śrubowe ARK, zworki, gniazdo goldpin i elektrolity. Już teraz proponuję włożenie układów scalonych w podstawki oraz, na samym końcu, zamontowanie stabilizatora U1. Jeśli pomiędzy płytkę sterownika oraz układu wykonawczego zdecydujemy się na użycie słupków dystansowych o wysokości 2 cm, wówczas stabilizator (najlepiej jeszcze



Rys. 5. Opis wyprowadzeń złącza programowania w systemie (ISP)

R E K L A M A

# STM32 FanClub

Pierwszy na świecie programator debugger dla STM32 w cenie 97 PLN

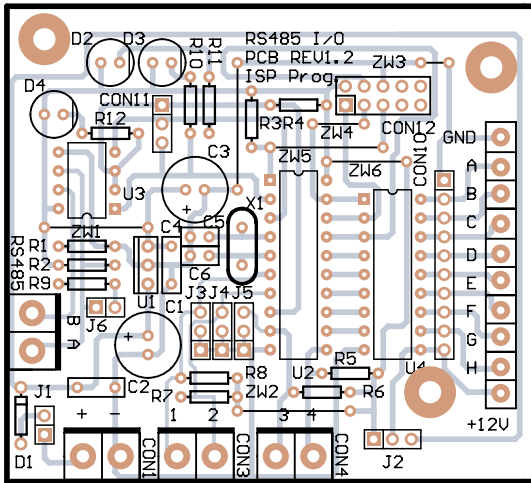
ZL30PRG

Dostępny m.in. w

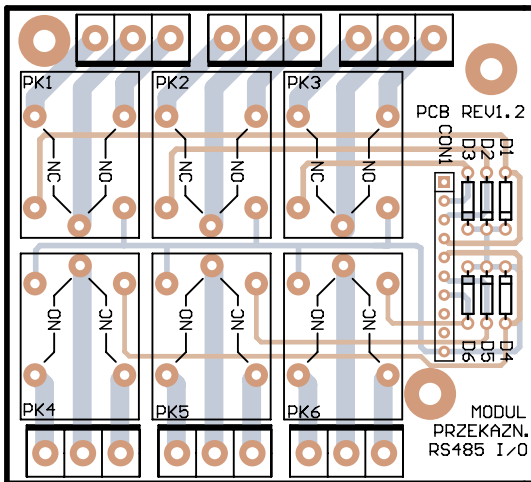
## KAMAMI

www.kamami.pl

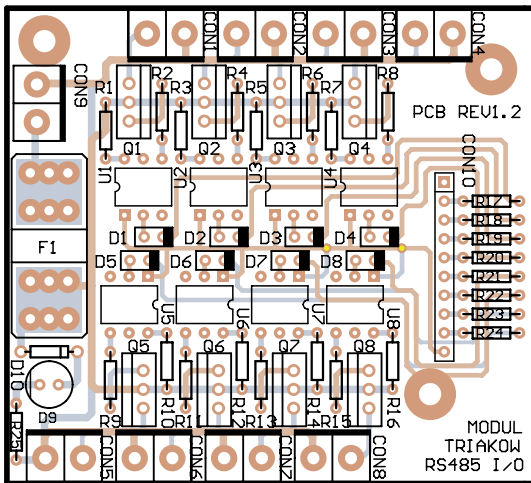




Rys. 6.



Rys. 7.



Rys. 8.

przed zamontowaniem) należy odpowiednio wyprofilować, zaginając jego wyprowadzenia na odległości ok 4 mm od podstawy i zamontować jak fot. 4.

Kolejnym krokiem jest odpowiednie ustawienie jumperów. W tym celu nale-

ży założyć jumper J1, jumper J2 ustawić w pozycji 2-3, za pomocą jumperów J3...J5 ustawić unikalny adres urządzenia, i, w razie potrzeby włączyć rezystor terminujący magistralę RS485 za pomocą jumpera J6. Teraz należy urządzenie zaprogramować. W tym celu do złącza CON9 (ISP) wpinamy kabel programatora i podłączamy zasilanie. Ponieważ istnieje kilka „niepisanych standardów” złącza ISP dla pewności załączam rysunek złącza – rys. 5.

Zapisując program w pamięci mikrokontrolera należy:

- ustawić źródło taktu zegarowego na zewnętrzny rezonator kwarcowy 8 MHz,
- odznaczyć (wylączyć domyślnie zaprogramowaną) opcję dzielenia sygnału zegarowego przez 8,
- zaprogramować mikrokontroler kodem programu wynikowego.

Po wykonaniu tych operacji i restarcie zasilania układ powinien zaświecić zieloną diodę LED D4 oznaczającą, że układ MAX485 pracuje w trybie odbioru, po czym na ułamek sekundy wygasić ją (MAX485 w trybie nadawania). W tym czasie mrugnie żółta dioda LED D2 oznaczająca aktywność na linii nadawczej (sterownik wysłał właśnie zgłoszenie do sieci, że jest aktywny). Oznacza to, z dużym prawdopodobieństwem, że układ został poprawnie zaprogramowany i działa prawidłowo.

**Moduł wykonawczy na przełącznikach.** Montaż układu wykonawczego z przełącznikami jest prosty i zajmuje zaledwie kilka chwil. I trudno się dziwić: wystarczy zamontować po sześć sztuk diod, złączy śrubowych ARK, przełączników (od strony elementów) oraz listwę goldpin (od strony lutowania – umownej bo płytkę jest dwustronna).

**Moduł wykonawczy na triakach.** Dla wygody, jak zwykle, jako pierwsze warto zamontować elementy najniższe: wszystkie rezystory, diodę D10, później podstawki pod optotriaki, wszystkie diody LED, złącza ARK CON1...CON9 i jako ostatnie po umownej stronie elementów

najwyższe – triaki. Po stronie lutowania, analogicznie jak w przypadku karty z przełącznikami, montujemy listwę goldpin, stanowiącą interfejs pomiędzy płytką układu wykonawczego i sterownika.

**Połączenie modułów w całość.** W zależności od potrzeby, uzbrajamy płytkę sterownika w płytkę układu wykonawczego opartego na przełącznikach lub triakach. W tym celu do spodu płytki układu wykonawczego, za pomocą śrubek przykręcamy trzy słupki dystansujące o wysokości minimum 2 cm. Rozstaw otworów na płytkach jest identyczny, dzięki czemu będziemy mogli tak przygotowaną płytkę układu wykonawczego nałożyć na płytkę sterownika (trafiając słupkami w otwory płytki sterownika), pilnując także by listwa goldpin układu wykonawczego trafiła w gniazdo na płycie sterownika. Tak zmontowane urządzenie jest już gotowe do pracy.

### Uruchomienie

Teoria komunikacji z urządzeniem, jak i sposób sterowania urządzeniami opisano podczas omawiania działania programu mikrokontrolera. Sterować kartami można np. za pomocą komputera PC ze złączem RS232 wyposażonego w konwerter RS232<->RS485.

Po podłączeniu urządzeń do magistrali RS485 (linia A konwertera z linią A karty wykonawczej, linia B konwertera z linią B karty wykonawczej) uruchamiamy dowolny program terminala (np. GtTerm pod Linuxem, albo Hyper Terminal pod Windows), wybieramy właściwy port COM i konfigurujemy parametry transmisji: szybkość transmisji 4800, bit kontroli parzystości: brak, liczba bitów danych: 8, liczba bitów stopu: 1 oraz kontrola przepływu: brak. Po wybraniu portu COM i prawidłowym skonfigurowaniu parametrów transmisji nie pozostaje już nic innego, jak tylko podłączyć zasilanie karty wykonawczej. Ta po chwili wyśle komunikat zgłoszenia, który powinien ukazać się w oknie Terminala. Teraz można już sterować podłączonymi do karty urządzeniami. Jeśli mimo wszystko ktoś będzie miał problem z obsługą transmisji szeregowej, konfiguracją programu terminala, zapraszam na bloga „Elektronika Po Godzinach” (<http://epg.saviportal.pl>) gdzie temat omówiłem szerzej i załączyłem stosowny instruktaż.

Mariusz Ciszewski  
mariusz.ciszewski@elportal.pl