

Moduł wejść cyfrowych z interfejsem RS-485

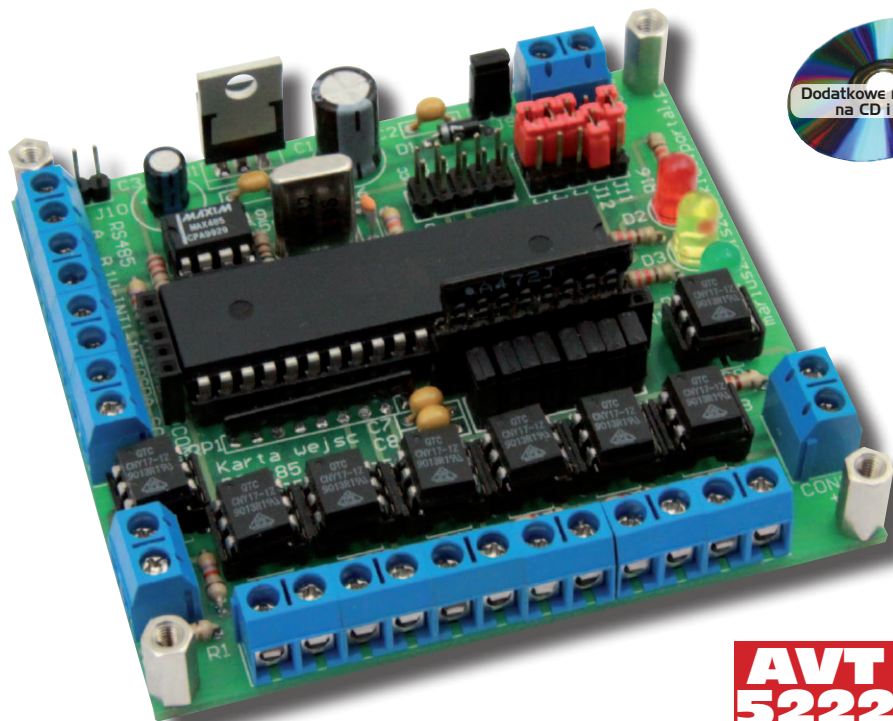
Prezentujemy kolejny projekt urządzenia przeznaczonego do zastosowania w systemach automatyki z akwizycją wielu danych. Jest to uniwersalny moduł wejść cyfrowych z interfejsem RS-485. Doskonały do zastosowania w systemach, gdzie zachodzi konieczność zbierania informacji z wielu urządzeń umieszczonych w znacznych odległościach od siebie. Moduł zbudowano w oparciu o mikrokontroler AVR i wyposażono w złącze do programowania mikrokontrolera w układzie. W podstawowej wersji ma 8 odizolowanych galwanicznie wejść, złącze 1Wire, I²C, INT0, INT1.

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

Podobnie, jak w przypadku opublikowanego wcześniej projektu modułu wykonawczego (EP 1/2010), tak i ten jest uniwersalny. Dlatego też jest możliwe zastosowanie modułu w dowolnych systemach wykorzystujących RS485. Przykładem takiego systemu sterowania może być komputer PC z konwerterem RS232 <-> RS485 (np. AVT530), z dołączonymi do magistrali wieloma modułami akwizycji danych, wykonawczymi, pełniącymi przeróżne zadania.

Opis urządzenia

Podobnie jak poprzednio, urządzenie zrealizowano w oparciu o popularny układ MAX485 o typowej dla standardu RS485 impedancji odbiornika równej 12 kΩ. Moduł w wersji podstawowej ma możliwość podłączenia dodatkowych układów, zwiększających jego funkcjonalność (np. liczbę oraz typ wejść). W wersji podstawowej, urządzenie ma osiem niezależnych, optoizolowanych wejść cyfrowych z opcją pamięci. Oznacza to, że urządzenie będzie mogło gromadzić informacje w postaci



**AVT
5222**

obecności lub braku napięcia z zakresu 5 V...12 V z ośmiu źródeł, które nie muszą być ze sobą w żaden sposób połączone elektrycznie. Optoizolacja minimalizuje też ryzyko ewentualnego sabotażu oraz niweluje wpływ wszelkiego rodzaju niekorzystnych zjawisk elektrycznych mogących pojawić się na monitorowanych przez kartę urządzeniach. Opcja pamięci może być ustawiona dla każdego z wejść indywidualnie i oznacza to, że, w przypadku jej ustawienia, urządzenie będzie pamiętało stan wystąpienia zdarzenia na danym wejściu tak długo, dopóki urządzenie sterujące (master) tej informacji nie anuluje. Takie rozwiązanie wyklucza możliwość, że master nie zdąży odczytać jakiejś ważnej informacji. W przeciwnym wypadku (gdy opcja pamięci dla danego wejścia będzie wyłączona), na zapytanie ze strony mastera o stan tego wejścia moduł poda stan chwilowy, obecny na wejściu tuż po odebraniu zapytania.

Schemat ideowy urządzenia pokazano na rys. 1. Znajdują się tu:

- Blok stabilizacji napięcia 5 V z układem 7805 (U1) i kondensatorami C1...C4.
- Blok sterujący z mikrokontrolerem ATmega16 (U2); wybrałem go nie tyle z po-

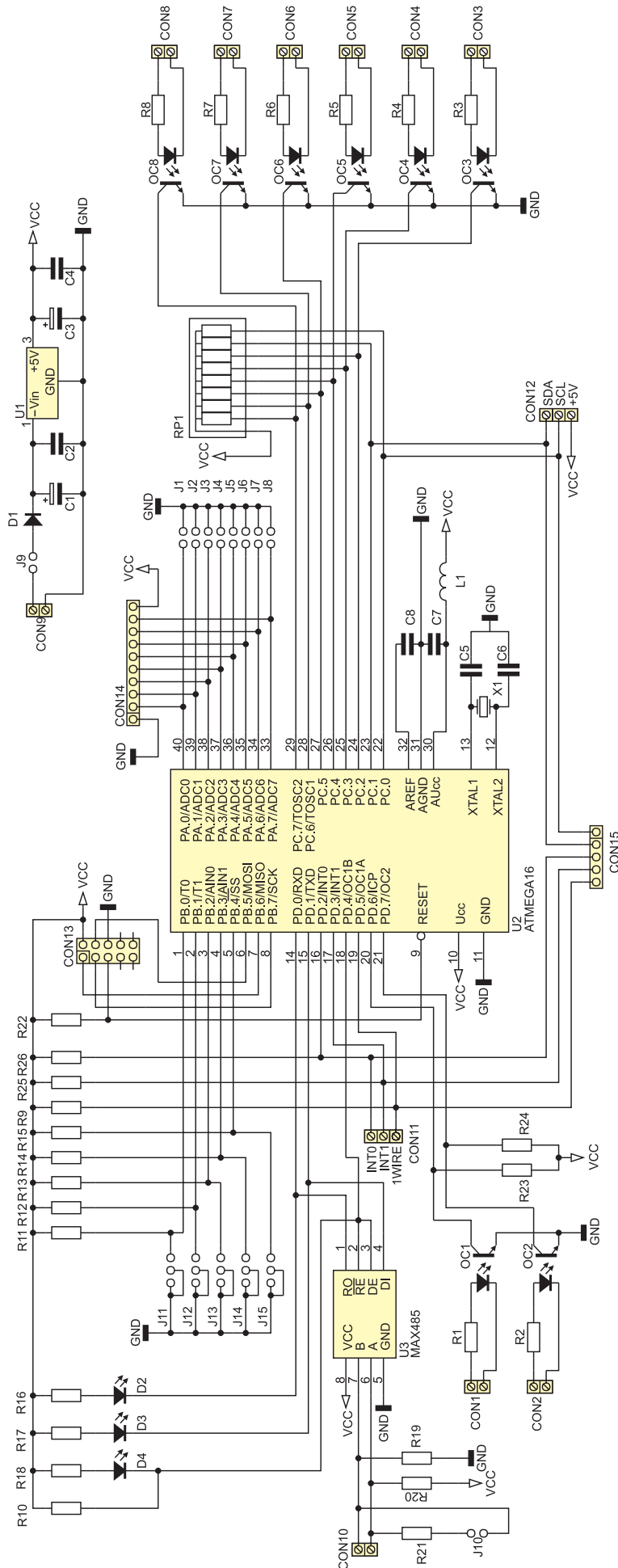
AVT-5222 w ofercie AVT:
AVT-5222A – płytką drukowaną
AVT-5222B – płytką drukowaną + elementy

Dodatkowe materiały na CD i FTP:
ftp://ep.com.pl, user: 12686, pass: 2b7r7b68
• wzory płytek PCB
• 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 optoizolacją (EP5/1999)
AVT-5006	Dwukierunkowy interfejs RS232/RS485 (EP4/2001)
AVT-5098	Konwerter USB – RS485 z separacją galwaniczną (EP2/2003)
AVT-530	Konwerter RS232 – RS485 (EP6/2003)
AVT-531	Karta wyjść przekaźnikowych (EP7/2003)
AVT-532	Karta wyjść optoizolowanych (EP7/2003)
AVT-533	Karta wyjść cyfrowych (aktywne GND) (EP8/2003)
AVT-534	Karta wyjść cyfrowych (aktywne VCC) (EP8/2003)
AVT-535	Karta wejść cyfrowych (EP9/2003)
AVT-536	Karta wejść analogowych (EP9/2003)
AVT-537	Moduł terminala z wyświetlaczem LED (EP10/2003)
AVT-538	Alfanumeryczny wyświetlacz LED (EP10/2003)
AVT-439	8-kanalowy przedłużacz analogowy z RS485 (EP1/2006)

wodu większej liczby wyprowadzeń, ale przede wszystkim ze względu na dużą, 16 kB pamięć flash, co zwiększyło moż-



Rys. 1. Schemat ideowy modułu

liwości implementacji różnych funkcji programu.

- Blok interfejsu z układem MAX485 (U3) dopasowującym sprzętowy interfejs szeregowy mikrokontrolera UART do wymagań RS485.

Złącze CON13 (ISP) pozwala „tchnąć życie” w nowo zbudowane urządzenie, a więc umożliwia programowanie mikrokontrolera bezpośrednio w układzie, bez potrzeby wyjmowania go z podstawki. Złącza CON1...CON8 są wejściami i służą do podłączenia zewnętrznych sygnałów wejściowych (napięcie 5...12 V jest interpretowane jako logiczna jedynka). Napięcie wejściowe ze złącz podawane jest za pośrednictwem rezystorów (R1...R8) na anody diod LED transoptorów (OC1...OC8). Wyjścia transoptorów podłączone są portów mikrokontrolera (PD6, PD7, PC...PC7), domyślnie podciągnięte za pomocą rezystorów zawartych w strukturze drabinki RP1 oraz R23, R24 do plusa zasilania, w stan niski. Na żądanie mastera informacje o stanie wejść CON1...CON8 są przekazywane w sieć za pośrednictwem złącza CON10. Rezystor R21 jest tzw. rezystorem terminującym, włączanym za pomocą zworki J10 wyłącznie w ostatnim (lub pierwszym i ostatnim) urządzeniu na linii transmisyjnej, w przypadku gdy zauważymy błędy transmisji, które mogą wynikać z ewentualnych odbić w linii transmisyjnej (z reguły nie ma takiej konieczności). Złącze CON11 wyprowadza bezpośrednio linie przerwań INT0, INT1 oraz PD5 przewidziane na realizację magistrali 1WIRE na zewnątrz układu. W wersji podstawowej nie są one obsługiwane przez firmware, jeśli jednak kiedyś pojawi się potrzeba ich implementacji nie będzie z tym większego problemu (wyprowadzone na zewnątrz złącza, pojemna pamięć programu). Podobnie złącze CON12 wyprowadza na zewnątrz linie przewidziane na realizację magistrali I²C oraz +5 V. Sygnały te dostępne są również w złączu CON15. Symbolami J1...J8 oznaczono jumpery służące do konfiguracji opcji pamięci dla poszczególnych wejść CON1...CON8. Założenie jumpera JX oznacza włączenie pamięci dla złącza CONX, gdzie X oznacza numer jumpera

R E K L A M A

STM32 FanClub
Sięgaj nieba...
Dla fanów STM32 mamy wszystko!

KAMAMI www.kamami.pl

jak i złącza. W przypadku chęci rozbudowy urządzenia (zastosowania modułu rozszerzeń) i wykorzystania portu A mikrokontrolera w sposób bardziej ambitny, niż tylko odczyt informacji na temat ustawienia zwerek, linie PA0...PA7 oraz zasilanie i masa zostały wyprowadzone na złącze CON14. Ze względu na dwojaką rolę portu A mikrokontrolera (obsługa jumperów lub złącze rozszerzeń) na płytce nie realizowano podciągania do plusa zasilania poszczególnych jego wyprowadzeń (co jest konieczne do prawidłowego odczytu ustawień jumperów przez mikrokontroler, ale miało by negatywny wpływ na ewentualny odczyt wartości analogowych przez mikrokontroler w przypadku zastosowania modułu rozszerzeń). Dlatego w przypadku chęci wykorzystania portu A do odczytu konfiguracji zwerek (a tak jest w podstawowej wersji układu) w złączu CON14 należy umieścić drabinkę rezystorową, zgodnie z opisem w punkcie „montaż i uruchomienie”. Oczywiście, jeżeli opcja pamięci ma być domyślnie włączona dla wszystkich wejść, należy założyć wszystkie jumpery J1...J8, wówczas stosowanie wspomnianej drabinki w złączu CON14 nie jest potrzebne. Złącza CON14 i CON15 są złączami rozszerzeń i stanowią drogę komunikacji urządzenia z opcjonalnymi modułami dodatkowymi (rozbudowa karty o kolejny poziom, dodanie nadrzędnej równoległej płytki PCB, analogicznie jak miało to miejsce w przypadku karty wykonawczej). Jumpery J11...J15 służą do ustawienia niepowtarzalnego adresu sieciowego karty (poprzez zwieranie do masy normalnie podciągniętych do +5 V za pomocą rezystorów R11...R15 wyprowadzeń PB0...PB4 mikrokontrolera). W celu ustawienia odpowiedniego adresu karty należy za pomocą tych jumperów ustawić binarną kombinację, odpowiadającą liczbie w zakresie 0...31. Zworka w pozycji 1-2 oznacza logiczne zero, w pozycji 2-3 logiczną jedynkę, przy czym J1 reprezentuje najmłodszy, a J5 najstarszy bit słowa adresu. Przykładowo, ustawienie wszystkich jumperów J1...J5 w pozycji 1-2 jest równoznaczne z nadaniem karcie adresu 0, a przestawienie ich wszystkich w pozycję 2-3 ustawi adres karty na 31. Diody LED: D3, D2 oraz D4, pozwalają 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 stan linii sterującej układu MAX485. Gdy D4 świeci, to MAX485 jest w trybie odbioru, a gdy zgaszona – nadawania.

Do złącza CON1 należy doprowadzić stałe napięcie zasilające z zakresu 8...12 V. Dioda D1 zabezpiecza układ przed uszkodzeniem w przypadku niepoprawnego podłączenia biegunów zasilania.

Oprogramowanie mikrokontrolera

Pierwszą operacją wykonywaną po starcie mikrokontrolera jest odczyt konfiguracji jumperów J11...J15 i ustawienie adresu sieciowego karty. Jest on wykonywany jednorazowo za pomocą operacji przypisania $Adres = Pinb \text{ And } \&B00011111$. Trzy najmłodsze bity rejestru *Pinb* są maskowane. Po ustaleniu swojego adresu sieciowego, karta odczeka bezpieczny, obliczony na podstawie adresu odcinek czasu, a następnie wyśle w sieć zgłoszenie swojej obecności. Wspomniany odcinek czasu pozwala uniknąć kolizji danych w przypadku, gdyby wszystkie urządzenia chciały zacząć nadawać w jednym czasie np. po powrocie napięcia zasilającego w linii, po wcześniejszej jego utracie. Dzięki takiemu rozwiązaniu, przy jednoczesnym załączeniu zasilania urządzenia powinny zgłaszać się kolejno, poczynając od tego o najniższym adresie, ze zwłoką minimum 2 sekund. Odpowiada za to podprogram:

```
Sub Czekaj
  Wait 2
  For Opoznienie$ = 1 To Adres
    Waitms 300
  Next Opoznienie$
End Sub
```

Następnie odczytywany jest stan jumperów J1...J8, czyli konfiguracja opcji pamięci dla poszczególnych wejść. Jeżeli dla któregośkolwiek wejścia opcja pamięci została załączona, zostaną uruchomione przerwania, a informacja o tym, wraz z numerami wejść dla których opcja pamięci została włączona, zostanie wysłana przez sieć. Przykładowo informacja tak może wyglądać następująco: *Urz: 11, FV-1.0: Opcja pamieci aktywna dla wejsc: 1, wlaczone przerwania.*

Następnie wysłany zostanie właściwy komunikat zgłoszeniowy w postaci ciągu znaków: *Urz: 11, FV-1.0: OK > [TKx1][NKx2][TWx1][NWx2]*. Identyfikuje on zgłaszające się urządzenie (adres sieciowy oraz wersja firmware) oraz ciąg znaków sugerujący oczekiwanie na słowo sterujące. Słowo sterujące może się składać maksymalnie z sześciu znaków, według klucza: *[TK][NK][TW][NW]*, gdzie:

- TK – typ karty – został stworzony do wyodrębnienia grupy urządzeń wewnątrz sieci. Dla kart wejściowych w jego miejsce należy stosować literę „i”, skrót od input. Dzięki jego wprowadzeniu pozostałe urządzenia w sieci, nie będąc kartami wejścia (np. karty wykonawcze z poprzedniego artykułu) nie będą reagowały na polecenia wydawane tej grupie urządzeń.
- NK – numer karty – adres sieciowy (liczba z przedziału 0...31), dzięki któremu na zadane żądanie ze strony mastera odpowie tylko jedna, właściwa karta wejść w sieci. Parametr należy podawać zawsze w postaci dwóch znaków, np. 01 dla karty o numerze (adresie) 1.

- TW – typ wejścia – identyfikator rodzaju wejścia. Ponieważ karta (uzbrojona w dodatkowy moduł rozszerzający) będzie miała na swoim pokładzie różne rodzaje wejść (analogowe, cyfrowe), jego podanie pozwoli stwierdzić do jakiej grupy wejść się odnosimy. W wersji podstawowej karty, bez modułu rozszerzeń (a więc wyposażoną w optoizolowane wejścia cyfrowe), należy używać w tym miejscu parametru „d” odpowiadającego słowu „digital”, czyli „cyfrowe”.
- NW – numer wejścia – liczba z przedziału 0...99 określająca, do którego wejścia z grupy TW odnosimy się w danej chwili (podstawowa wersja karty ma ich 8). Parametr należy podawać zawsze w postaci dwóch znaków, np. 01 dla wejścia o numerze 1.

Po wyjaśnieniu powyższych skrótów obsługa karty powinna stać się intuicyjna. Poniżej kilka przykładów.

Odczyt stanu pojedynczego wejścia. Jeżeli chcemy odczytać stan panujący na pierwszym wejściu cyfrowym karty wejść o adresie „1”, wówczas na komunikat zgłoszeniowy w postaci *[TK][NK][TW][NW]* odpowiemy 6 znakowym słowem *i01d01* – co czytamy: karta wejść (i) numer 1 (01), wejścia cyfrowe (d), numer wejścia 1 (01). Karta zwróci wówczas (wyśle w sieć) odpowiedź, przykładowo w postaci: *Urz: 11, FV-1.0: OK > Ans(d1)=0*, informujący o tym, że na pierwszym wejściu cyfrowym karty w tym konkretnym przypadku panuje stan niski (czyli, że na złączu CON1 karty nie jest podawane napięcie).

Odczyt stanu wszystkich wejść danego typu. W praktycznym użytkowaniu karty, może się okazać, że w pewnych sytuacjach wygodniej będzie zapytać kartę o stan wszystkich wejść cyfrowych na karcie jednocześnie. W takim przypadku wystarczy użyć słowa 4 znakowego, z pominięciem numeru konkretnego wejścia. Takim 4-znakowym zapytaniem może być słowo *i01d*. Przykładowa odpowiedź na takie zapytanie przyjmie postać: *Urz: 11, FV-1.0: OK > Ans(d1...d8)=0,0,0,0,0,0,0,0*. Widać, że w tym przypadku wszystkie cyfrowe wejścia karty miały stan logicznego zera.

Kasowanie pamięci dla pojedynczego wejścia cyfrowego. Jeżeli dla danego wejścia została włączona opcja pamięci (jumpery J1...J8), wówczas, jeśli wystąpiło na nim zdarzenie, karta pamięta je tak długo, aż pamięć o tym zdarzeniu nie zostanie wykasowana. Tym samym, ponowne sprawdzanie stanu danego wyprowadzenia, które choć raz weszło w stan aktywny będzie miało sens dopiero po wcześniejszym wykasowaniu zapamiętanego stanu aktywnego. W tym celu wystarczy zastosować słowo 6 znakowe, przy czym w miejsce znacznika *[TW]* należy użyć znaku „c” (od słowa „clear”). Teoretycznie pominęliśmy ważną informację o tym, że chodzi o wejścia cyfrowe, jednak ponieważ opcja pamięci dotyczy wyłącznie wejść cyfrowych, nie ma to większego znaczenia.

Wykaz elementów

Rezystory:

R1...R8: 820 Ω
 R9...R15, R22...R26: 4,7 kΩ
 R16...R18: 220 Ω
 R19, R20: 510 Ω
 R21: 120 Ω
 RP1: drabinka 8×4,7 kΩ

Kondensatory:

C1: 470 μF/16 V
 C2, C4, C7, C8: 100 nF
 C3: 47 μF/16 V
 C5, C6: 33 pF

Półprzewodniki:

U1: 7805 (TO-220)
 U2: ATmega16 (DIP40)
 U3: MAX485 (DIP8)
 D1: 1N4001
 D2: LED 5 mm (czerwona)
 D3: LED 5 mm (żółta)
 D4: LED 5 mm (zielona)
 OC1...OC8: CNY17

Inne:

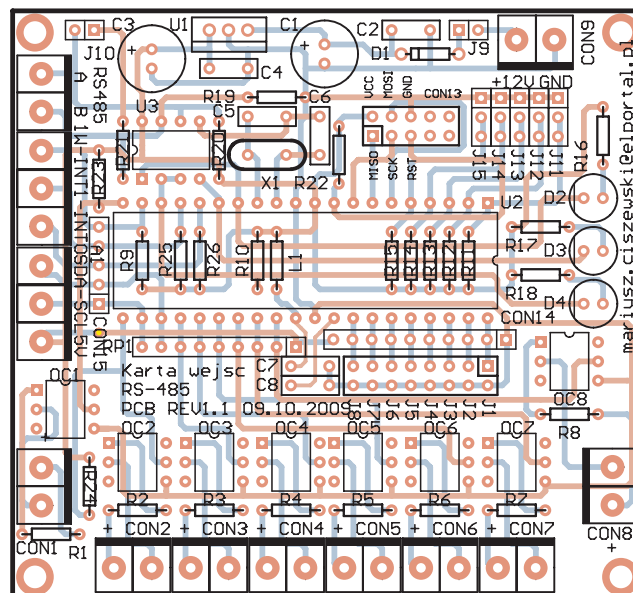
X1: kwarc 1,8432 MHz
 CON1...CON10: ARK2 (5 mm)
 CON11, CON12: ARK3 (5 mm)
 CON13: goldpin 5×2
 CON14: goldpin 10×1
 CON15: goldpin 5×1
 J1...J8: zworki+goldpin 8×2
 J9, J10: zworki+goldpin 2×1
 J11...J15: goldpin 5×1
 Podstawki pod układy scalone i transoptory
 L1: dławik 10 μH
 Dodatkowo: drabinka rezystorowa 8×47 kΩ+goldpin 9×1

nego wejścia bez przestawiania zwerek J1...J8 bez ponownego uruchomienia urządzenia. Aby włączyć opcję pamięci dla wybranego wejścia po sieci RS485 wystarczy wysłać do urządzenia słowo 6-znakowe w postaci [TK][NK][TW] [NW], przy czym jako [TW] użyć znaku „m” (od „memory”).

Programowe załączanie opcji pamięci dla wszystkich wejść cyfrowych. Aby operację tę wykonać dla wszystkich wejść jednocześnie, wystarczy użyć słowa 4-znakowego [TK][NK][TW] [NW] (pomijamy znacznik [NW]), i podobnie jak wcześniej jako [TW] używamy znaku „m” (od „memory”).

Programowe wyłączenie opcji pamięci dla pojedynczego wejścia cyfrowego. Aby w trakcie działania karty zmienić tryb pracy dla wybranego wejścia (włączyć opcję pamięci) bez przestawiania zwerek J1...J8 oraz bez ponownego uruchomienia urządzenia, wystarczy postąpić podobnie jak w opisano wcześniej, tym razem jako [TW] używając znaku „n” (od „no memory”). Uwaga! Polecenie nie resetuje zapamiętanego stanu. Po ponownym programowym włączeniu opcji pamięci wciąż pamiętany będzie stan sprzed jej wyłączenia.

Programowe wyłączenie opcji pamięci dla wszystkich wejść cyfrowych. Aby operację tę wykonać dla wszystkich wejść jednocześnie, wystarczy użyć słowa 4-znakowego [TK][NK][TW] (pomijamy znacznik [NW]), i podobnie jak wyżej jako [TW] używamy znaku „n” (od „no memory”). Uwaga! Polecenie nie zmienia zapamiętanych stanów. Po ponownym programowym włączeniu opcji pamięci wciąż pamiętane będą stany sprzed jej wyłączenia.



Rys. 2. Schemat montażowy modułu

Zaraz po odebraniu słowa sterującego z sieci RS485 karta dokonuje jego parsowania i przypisuje odpowiednie wartości poszczególnej zmiennej. Odpowiada za to następujący fragment programu:

```
Typ_karty$ = Mid(rozkaz , 1 , 1 )
Zmienna_interpretacyjna =
Mid(rozkaz , 2 , 2)
Numer_karty$ = Val(zmienna_
interpretacyjna)
Typ_wejscia$ = Mid(rozkaz , 4 , 1 )
Zmienna_interpretacyjna =
Mid(rozkaz , 5 , 2)
Numer_wejscia$ = Val(zmienna_
interpretacyjna)
```

W następnej kolejności zostaje sprawdzona liczba znaków odebranego ciągu znaków $Dlugosc_slova\$ = Len(rozkaz)$.

Na podstawie długości odebranego ciągu znaków oraz sprawdzeniu odpowiednich zależności z wykorzystaniem powyższych zmiennych następuje realizacja odpowiedniego polecenia.

Po wykonaniu tych operacji program wraca do początku (czeka na kolejne wpro-

Kasowanie pamięci dla wszystkich wejść cyfrowych



Może się okazać, że w pewnych sytuacjach (np. ponowne uruchomienie systemu) wygodne okaże się kasowanie pamięci dla wszystkich wejść za pomocą jednej komendy. By osiągnąć taki efekt wystarczy użyć słowa 4-znakowego w postaci [TK][NK][TW], przy czym, podobnie jak wyżej, znacznik [TW] należy zastąpić znakiem „c” (od słowa „clear”).

Programowe załączanie opcji pamięci dla pojedynczego wejścia cyfrowego. Można w trakcie działania karty (np. w celach diagnostycznych) zmienić tryb pracy dla wybra-

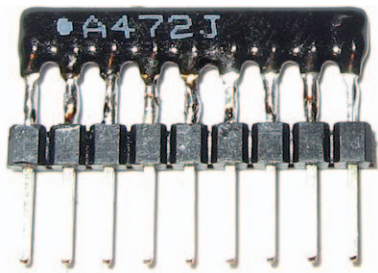
R E K L A M M A

autoryzowany dystrybutor
www.micros.com.pl

- ⊙ akumulatory NiMH i NiCd
- ⊙ baterie alkaliczne
- ⊙ baterie litowe
- ⊙ baterie cynkowo-chlorkowe
- ⊙ na specjalne życzenie klienta dostarczamy niestandardowe produkty

MICROS Sp. j.
 Kraków, ul. Godlewskiego 38
 tel. 12 636 95 66
 biuro@micros.com.pl



Fot. 3. Drabinka rezystorowa gotowa do zamontowania w złączu CON14

wadzone rozkazy i ewentualnie wykonuje je, jeśli są adresowane do niego, bądź ignoruje, jeśli kierowane są do innych urządzeń).

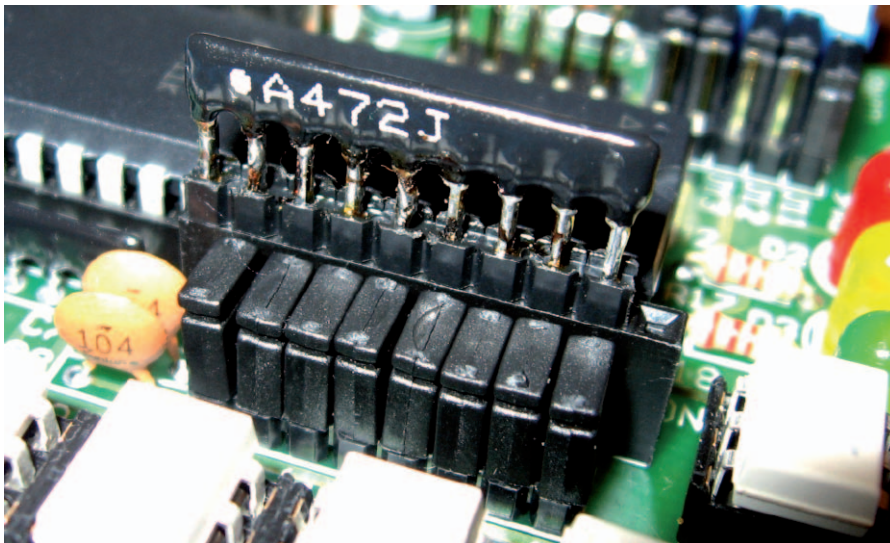
Obsługa przerwania

Jeśli dla przynajmniej jednego wejścia została włączona opcja pamięci, wówczas po włączeniu urządzenia uruchomione zostaną przerwania. Od tej pory, niezależnie od tego na jakim etapie wykonania znajduje się program główny, dwa razy na sekundę odbywa się sprawdzenie stanu wejść i w przypadku wykrycia stanu aktywnego fakt ten zostanie zapamiętany aż do wyraźnego polecenia wykasowania informacji o zdarzeniu przez master. Aby skonfigurować przerwanie w taki sposób by było generowane dwa razy w ciągu sekundy wystarczy:

- Obliczyć czas trwania pojedynczego cyklu maszynowego. Sprawdzić, ile takich cykli potrzeba do uzyskania potrzebnego czasu, przy założeniu, że wykorzystany zostanie licznik 16 bitowy, upewnić się, że liczba potrzebnych cykli nie przekroczy wartości 65535. W przeciwnym wypadku zastosować optymalny podział częstotliwości sygnału wejściowego i obliczyć liczbę potrzebnych cykli z jego uwzględnieniem.
- Skonfigurować timer oraz - w celu ustawienia wartości do jakiej ma zliczać licznik w celu wygenerowania przerwania - wpisać do jego rejestru wartość będącą różnicą 65535 oraz liczby potrzebnych cykli.
- Uruchomić timer i przerwania.

W urządzeniu zastosowano rezonator kwarcowy o częstotliwości drgań 1,8432 MHz, stąd okres wynosi 542,5 μs. Dla uzyskania odcinków czasu 0,5 s należałoby zliczyć 921600 cykli zegarowych. Przekracza to jednak możliwości 16-bitowego timera, stąd niezbędne jest zastosowanie preskalertera. Przy zastosowaniu podziału przez 64 dla uzyskania 0,5 s odcinka czasu wystarczy zliczyć już tylko 14400 cykli. Różnica pomiędzy 65535 a 14400 wynosi 51135. Pozostaje skonfigurować timer:

```
Config Timer1 = Timer , Prescale
= 64
On Timer1 Przerwanie
Timer1 = 51135
Enable Interrupts
Enable Timer1
```



Fot. 4. Prawidłowy sposób zamontowania drabinki w złączu CON14

Od tej pory, niezależnie od stanu wykonywania się programu głównego dwa razy w ciągu jednej sekundy wykonywana zostaje zawartość procedury *Przerwanie*.

Montaż i uruchomienie

Schemat montażowy modułu pokazano na rys. 2. Układ zmontowano na płycie dwustronnej z metalizacją. Elementy montuje się od najniższych do najwyższych. Będą to odpowiednio wszystkie rezystory, dioda D1, dławik L1, następnie podstawki pod układy scalone (w tym pod transoptory), drabinka rezystorowa RP1, kondensatory ceramiczne, złącza CON14 i CON15, wszystkie goldpiny, diody LED D2...D4, złącza śrubowe CON1...CON12, kondensatory elektrolityczne, rezonator kwarcowy X1 i na końcu najwyższy - stabilizator U1. Już na tym etapie warto założyć wszystkie potrzebne jumpery, ustawić adres sieciowy karty za pomocą jumperów J11...J15, ustawić opcję pamięci dla pożądaných wejść poprzez założenie jumperów J1...J8 oraz założyć jumper J9 załączający napięcie zasilania.

Po sprawdzeniu poprawności montażu należy włączyć napięcie zasilające i przed włożeniem układów w podstawki zmierzyć, czy na odpowiednich pinach podstawek pojawiają się oczekiwane napięcia zasilania i masy. Jeśli wszystko zgadza się, to można zamontować układy scalone. Czynność ta kończy montaż.

Teraz należy zaprogramować mikrokontroler. W tym celu do złącza CON13 (ISP) wpinamy kabel programatora i podłączamy zasilanie. Oprócz zapisu do pamięci flash mikrokontrolera samego programu, niezmiernie ważne jest odpowiednie ustawienie bitów Fusebits. Przede wszystkim należy wyłączyć interfejs JTAG oraz ustawić źródło taktu zegarowego na zewnętrzny rezonator kwarcowy małej częstotliwości.

Po zaprogramowaniu mikrokontrolera i ponownym uruchomieniu układu, powinna zaświecić się zielona dioda D4 informująca,

że układ MAX485 pracuje w trybie odbioru. Następnie powinna ona na ułamek sekundy zgasnąć, co sygnalizuje wysłanie opisanego wcześniej komunikatu sieciowego. W tym czasie mrugnie też żółta dioda LED D3 oznaczająca aktywność na linii nadawczej.

Jak wspomniano wcześniej, aby port A mikrokontrolera mógł prawidłowo odczytywać konfigurację jumperów J1...J8 służących do sprzętowego załączania opcji pamięci dla poszczególnych wejść w złączu CON14 należy umieścić drabinkę rezystorową 8×4,7 kΩ. Propozycję montażu drabinki przedstawiono na fot. 3 i fot. 4.

Komunikacja z kartą i sterownie urządzeniami

Teoria komunikacji z urządzeniem, jak i sposób sterowania urządzeniami (rozkazy sterujące) przedstawiono podczas omawiania działania programu mikrokontrolera. Sterować modułami można na przykład za pomocą komputera PC ze złączem RS232 i konwerterem RS232<->RS485 (dostępny w sklepie AVT np. jako kit AVT530). Po podłączeniu do linii RS485 urządzeń (linia A konwertera z linią A karty wykonawczej, linia B konwertera z linią B karty wykonawczej) uruchamiamy dowolny program terminala (np. GtKTerm pod Linuxem, albo Hyper Terminal pod Windows), wybieramy właściwy port COM i konfigurujemy parametry transmisji: szybkość transmisji 4800 bps, parzystość: brak, bity danych: 8, 1 bit stopu, brak kontrola przepływu. Po włączeniu zasilania modułu, po chwili powinien on wysłać komunikat zgłoszenia, który zostanie pokazany w oknie terminala. Jeśli pomimo wskazówek będą problemy z obsługą transmisji szeregowej, konfiguracją programu terminala, zapraszam do odwiedzenia strony „Elektronika Po Godzinach” (<http://epg.saviportal.pl>).

Mariusz Ciszewski
mariusz.ciszewski@elportal.pl