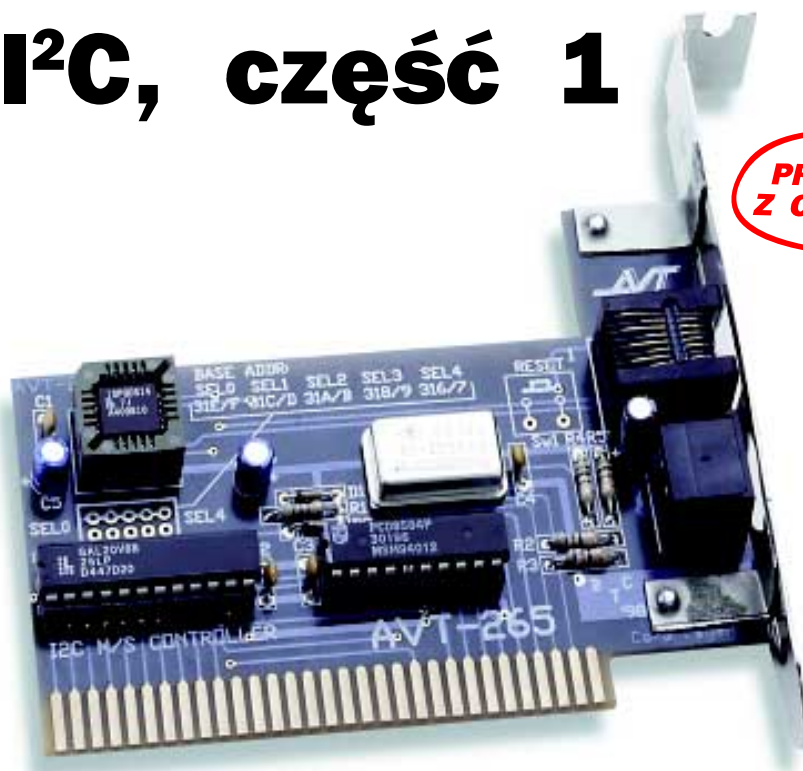


# Interfejs I<sup>2</sup>C, część 1

## kit AVT-265

PROJEKT  
Z OKŁADKI



Zagadnienia związane z interfejsem I<sup>2</sup>C cieszą się ogromną popularnością wśród naszych Czytelników. Wychodząc naprzeciw żądaniom zgłaszanym listownie i poprzez internetową „Listę Dyskusyjną”, przedstawiamy opis konstrukcji uniwersalnego, programowalnego interfejsu I<sup>2</sup>C, który umożliwi pracę w każdym trybie dopuszczalnym przez normy opisujące standard.

Stosunkowo niewielka liczba zmian w specyfikacji standardu I<sup>2</sup>C od chwili wprowadzenia go na rynek (1981 rok), świadczy o doskonałości założeń poczynionych na początku jego tworzenia. Macie więc teraz szansę dobrze poznać jedno z najbardziej klasycznych opracowań współczesnej elektroniki, stosowane zarówno w sprzęcie popularnym, jak i w szeregu aplikacji profesjonalnych.

Istnieje bardzo wiele możliwości wykonania interfejsu I<sup>2</sup>C dla komputerów PC. W najprostszyc realizacjach wystarcza jeden układ 7406 i kilka rezystorów, a najbardziej złożone wymagają zastosowania mikrokontrolerów sterowanych rozbudowanym oprogramowaniem. My wybraliśmy rozwiązanie pośrednie - wszystkie zadania związane z konwersją danych z postaci równoległej na szeregową i synchronizację transmisji wykonuje specjalizowany układ firmy Philips. Nosi on oznaczenie PCF8584. Za inicjalizację tego układu oraz przesyłanie danych pomiędzy użytkownikiem i rejestrami układu PCF8584, odpowiada prosty program, który steruje pracą komputera PC.

Zanim przejdziemy do omawiania tajników konstrukcji interfejsu, pokrótce przypomnimy po co wymyślono I<sup>2</sup>C i w jaki sposób następuje wymiana informacji pomiędzy układami dołączonymi do magistrali.

### Bity, ramki, warunki...

Na rys. 1 przedstawiono schemat blokowy systemu mikroprocesorowego, w którym wymiana informacji pomiędzy wszystkimi układami odbywa się dwoma przewodami! W ten sposób, już na samym początku artykułu, znaleź-

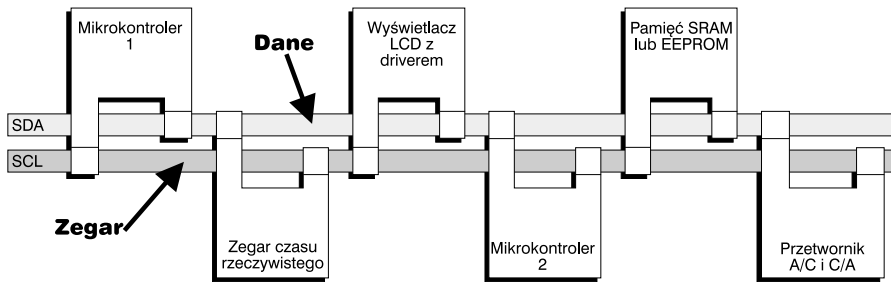
liśmy odpowiedź na pytanie: po co wymyślono ten interfejs? Chodziło przede wszystkim o zminimalizowanie liczby przewodów (linii) łączących układy scalone w dużych systemach sterowania.

Ograniczenie liczby linii, oprócz uproszczenia połączeń na płycie drukowanej, umożliwia zminimalizowanie zakłóceń elektromagnetycznych generowanych przez system mikroprocesorowy. Obydwie wymienione cechy mają ogromne znaczenie przede wszystkim w sprzęcie audio, video i telefonicznym, a właśnie z myślą o tego typu urządzeniach powstał I<sup>2</sup>C.

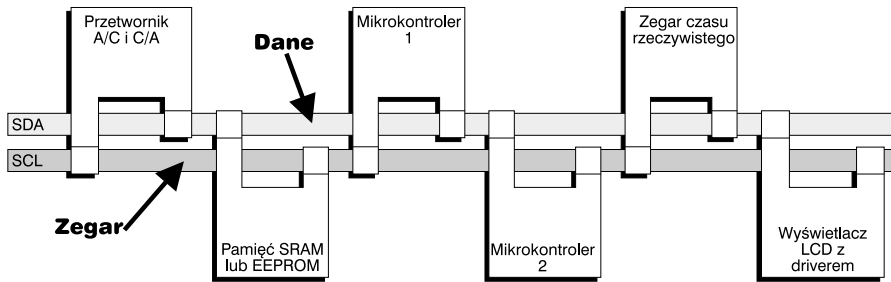
Już na pierwszy rzut oka można zauważyć, że sposób wymiany informacji pomiędzy elementami systemu ogranicza szybkość jej przepływu. Wynika to, po pierwsze, z charakteru przesyłania sze-

#### Cechy charakterystyczne, wymagania i możliwości interfejsu AVT-265.

- ✓ wykonany jest w postaci 8-bitowej karty do PC;
- ✓ współpracuje z dowolnym komputerem PC;
- ✓ rejestry interfejsu zajmują dwa adresy w przestrzeni I/O;
- ✓ istnieje możliwość wybrania jednej spośród pięciu par adresów (316/7h, 318/9h, 31A/Bh, 31C/Dh, 31E/Fh);
- ✓ może pracować w trybie MASTER, SLAVE lub MONITOR;
- ✓ jest w pełni programowalny, dzięki czemu bez większego trudu można go dostosować do indywidualnych wymagań użytkownika;
- ✓ magistrala może być taktowana jedną z czterech częstotliwości: 1,5kHz, 11kHz, 45kHz, 90kHz.



Rys. 1. Przykładowy system z magistralą I<sup>2</sup>C.



Rys. 2. Przykład innego systemu z magistralą I<sup>2</sup>C.

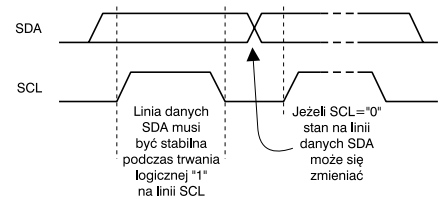
regowego (bit po bicie), a po drugie, z negatywnego wpływu pojemności i indukcyjności pasywnych, które zniekształcają przesyłany sygnał (zwłaszcza przy transmisji na odległości większe niż kilkadziesiąt centymetrów). Wada ta nie ujawnia się w typowych aplikacjach I<sup>2</sup>C, czyli w systemach sterowania i regulacji w odbiornikach telewizyjnych, sprzęcie elektroakustycznym, systemach nadzorczych w nowoczesnych komputerach PC i temu podobnych. Dla aplikacji wymagających większych szybkości transferu, Philips opracował rozszerzenie standardu I<sup>2</sup>C, które pokrótce omówimy w dalszej części artykułu.

Kolejną zaletą interfejsu I<sup>2</sup>C jest możliwość prostej rekonfiguracji „sprzętu” wchodzącego w skład systemu. Oznacza to, że konieczność rozszerzenia systemu z rys. 1 np. o kolejny sterownik wyświetlacza LCD i przetwornik C/A nie wymaga żadnej jego przeróbki! Jedyną konieczną modyfikacją jest taka

zmiana programu sterującego procesorem, aby adresował nowe układy i nie adresował np. zdemonstrowanego układu. Inną możliwością jest automatyczne diagnozowanie przez procesor, czy adresowany układ jest w danej chwili podłączony do szyny (jak to zrobić opowiemy za chwilę). Należy pamiętać, że adresowanie układu, który nie jest włączony w system nie zakłóca pracy pozostałych układów.

Kolejną, bardzo interesującą cechą I<sup>2</sup>C jest fakt, że nie jest istotne miejsce włączenia poszczególnych układów w szynę I<sup>2</sup>C. Z tego wynika, że systemy z rys. 1 i rys. 2 są z punktu widzenia I<sup>2</sup>C identyczne!

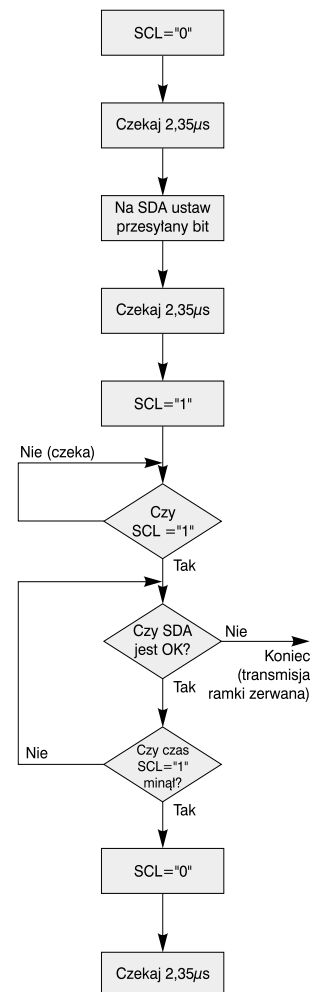
Ponieważ do szyny I<sup>2</sup>C można dołączyć wiele układów jednocześnie, niezbędne jest zastosowanie odpowiedniego protokołu, który zapewni poprawną wymianę informacji pomiędzy nimi. Kłopot jest o tyle duży, że w jednym systemie mogą się pojawić zarówno układy wyjściowe (informacja do nich jest tylko zapisywana, np. sterownik wyświetlacza LCD lub LED), wejściowe (np. dekodery klawiatur) oraz dwukierunkowe (zapis i odczyt danych, np. pamięci, przetworniki, czujniki temperatury). Co więcej, jak widać na rys. 1 i rys. 2, w jeden system można włączyć dwa (lub więcej) procesory! Zagadnienie jest naprawdę złożone, lecz inżyniero-



Rys. 3. Sposób przesłania bitu magistralą I<sup>2</sup>C.

wie Philipsa poradzili sobie z nim doskonale.

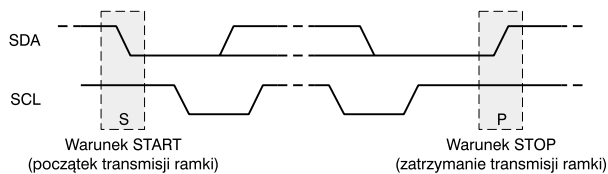
Warunkiem pracy dowolnego systemu z szyną I<sup>2</sup>C jest zastosowanie w nim minimum dwóch układów: jeden z nich spełnia rolę głównego układu sterującego i synchronizującego transmisję (Master), a drugi jest pomocniczy, zgodnie z angielską nomenklaturą nazywany niewolnikiem (Slave). Niezależnie od tego, czy dany układ spełnia rolę Mastera, czy też Slave'a, informacja może być przesyłana zarówno „do” niego, jak i „z” niego. Różnica pomiędzy Masterem i Slavem jest tylko



Rys. 4. Algorytm przesłania bitu magistralą I<sup>2</sup>C.

**Cechy charakterystyczne magistrali I<sup>2</sup>C.**

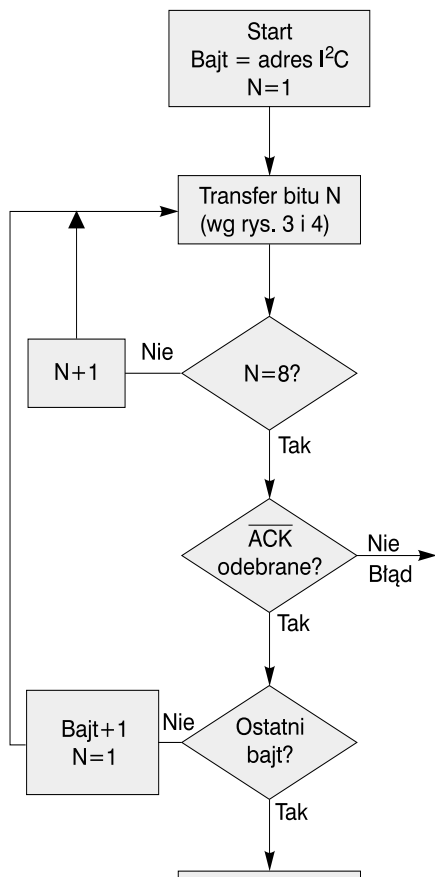
- ✓ przesyłanie 8-bitowych danych przy pomocy dwóch linii sygnałowych;
- ✓ łatwość modyfikacji sprzętowej urządzenia wykorzystującego szynę I<sup>2</sup>C;
- ✓ jego pracą steruje przejrzysty prosty protokół, pozwalający na jego łatwą implementację sprzętową lub programową;
- ✓ istnieje możliwość połączenia w jednym systemie kilku procesorów sterujących;
- ✓ możliwość generowania przerwań sprzętowych przez urządzenia dołączone do magistrali.



Rys. 5. Ramki START i STOP synchronizujące transmisję danych.

taka, że *Master* zawsze inicjuje i kończy transmisję, a także wyznacza jej tempo. Układy *Slave* są zawsze adresowane przez *Mastera*, przez co może on wpływać na to, który z układów będzie w danej chwili „dopuszczony” do magistrali I<sup>2</sup>C.

Podstawowym fragmentem informacji przesyłanej po szynie I<sup>2</sup>C jest bit. Na rys. 3 pokazano, w jaki sposób jest to robione. Podczas trwania logicznej „1” na linii zegarowej SCL, stan linii danych SDA nie może się zmieniać. Na rys. 4 przedstawiono algorytm transmisji pojedynczego bitu danych. Można go zastosować dla każdego mikrokontrolera, należy tylko samodzielnie zaprojektować opóźnienia wynikające z wymogów standardu.



Rys. 6. Algorytm przedstawiający sposób przesłania bajtu danych.

Nierozwiązany nadal pozostaje problem synchronizacji przesyłanego bloku informacji. Jednoznaczne wskazanie początku i końca takiego bloku odbywa się po-

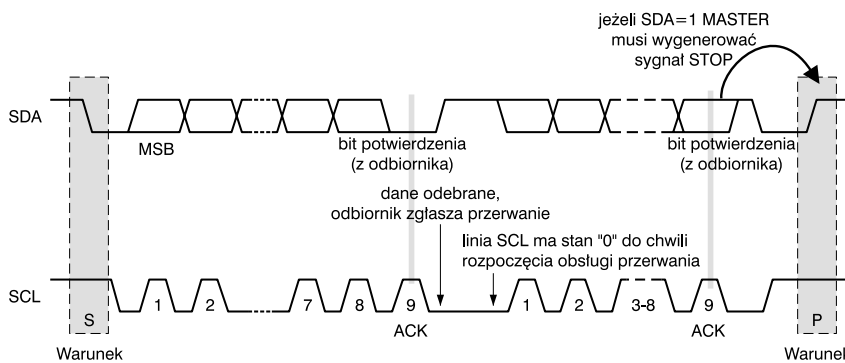
przez wygenerowanie przez *Mastera* sygnałów początku (*START*) i końca (*STOP*) transmisji. Sposób ich przesłania przedstawiono na rys. 5. Algorytm z rys. 6 przedstawia zalecany sposób przesyłania każdego bajtu.

Sygnaly *START* i *STOP* wyznaczają przesłanie kompletnego bloku informacji. Ponieważ transmisja po szynie I<sup>2</sup>C jest z założenia 8-bitowa, konieczne jest synchronizowanie przesłania każdego bajtu. Odbywa się to na drodze potwierdzenia odebrania bajtu przez układ podrzędny (*Slave*), przy pomocy sygnału ACK (z ang. ACKNOWLEDGE, czyli potwierdzenie). Aby uzyskać od *Slave'a* sygnał potwierdzenia, układ *Master* musi wygenerować kolejny, dziewiąty dla każdego bajtu impuls zegarowy. W tym czasie na linii danych SDA musi pojawić się stan logicznego „0” generowany przez *Slave'a* (*MASTER* odłącza się od linii SDA, poprzez ustawienie bufora wyjściowego w stan „1”). Brak potwierdzenia od strony *Slave'a* powoduje, że *Master* generuje sygnał *STOP*. Przebieg całego cyklu przesłania dwóch bajtów przedstawiono na rys. 7. Na rys. 8 przedstawiono najczęściej spotykaną ramkę danych przesyłaną poprzez I<sup>2</sup>C. Każda transmisja jest inicjowana przez *Mastera* i rozpoczyna się od wygenerowania sygnału *START*. Następnie przesyłany jest (także przez *Mastera*) 7-bitowy adres układu *Slave*. Przy pomocy

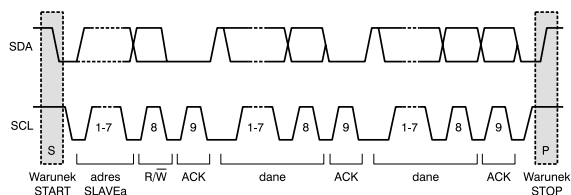
ósmego bitu słowa adresowego sygnalizowany jest kierunek przesyłania informacji - jeżeli jest on równy „1” to *Master* będzie odczytywał dane ze *Slave'a*, a w przypadku gdy ósmy bit jest równy „0”, *Master* będzie wpisywał kolejne bajty do *Slave'a*. Liczba bajtów przesyłanych do *Slave'a* jest dowolna (zależna od rodzaju odbiornika, w praktyce od typu układu), a koniec transmisji sygnalizowany jest warunkiem *STOP*.

W dość interesujący sposób rozwiązano interfejsy wejściowo-wyjściowe w układach I<sup>2</sup>C. Obydwie linie SDA i SCL są zazwyczaj dwukierunkowe, tzn. mogą pracować jako wejście lub wyjście (rys. 9). Jako bufony wyjściowe zastosowano tranzystory z otwartym drenem (w układach bipolarnych są to tranzystory z otwartym kolektorem). Dzięki zastosowaniu takiego rozwiązania możliwe jest jednoczesne podłączenie wielu układów do jednej linii magistrali.

Dla poprawnej pracy interfejsu I<sup>2</sup>C niezbędne jest „podwieszenie” linii SCL i SDA do plusa zasilania (w inny sposób nie jest możliwe uzyskanie napięcia odpowiadającego logicznej „1”). Wartość rezystancji rezystorów „podwieszających” należy dobrać w zależności od fizycznej długości magistrali, pojemności pasożytniczych (w skład których należy wliczyć także pojemności wejściowe układów dołączonych do magistrali) i żądanej szybkości transmisji. Na rys. 10 przedstawiono wykres obrazujący zależność pomiędzy pojemnością pomiędzy liniami SCL i SDA i maksymalną wartością rezystancji rezystorów „podwieszających”. Szybkość transmisji dla przedstawionego przykładu nie może przekraczać standardowo 100kHz.



Rys. 7. Wykres czasowy przedstawiający transfer dwóch bajtów danych.



Rys. 8. Format typowej ramki przesyłanej magistralą I<sup>2</sup>C.

Na tym zakończymy opis przybliżający podstawowe zasady przesyłania sygnałów z wykorzystaniem I<sup>2</sup>C. Rozszerzenia i udoskonalenia tego standardu, a także sposób arbitrażu w systemach z kilkoma *Masterami* przedstawimy w ostatniej części artykułu.

### Opis układu

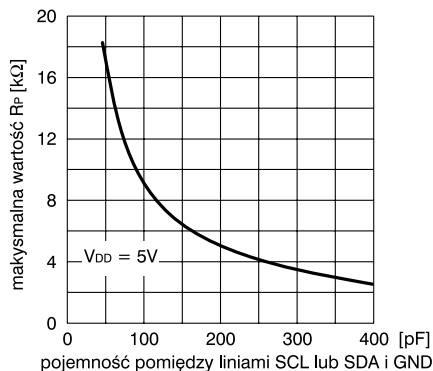
Przechodzimy teraz do omówienia konstrukcji interfejsu. Schemat elektryczny proponowanego rozwiązania znajduje się na rys. 11. Jak widać, jest to urządzenie bardzo proste, co było możliwe do osiągnięcia dzięki zastosowaniu dwóch nowoczesnych układów scalonych.

Układ programowalny US1 (GAL20V8) spełnia rolę dekodera adresowego, który pozwala dość swobodnie ustalić parę adresów, pod którymi interfejs będzie widoczny w przestrzeni I/O komputera PC. Na list. 1 przedstawiony został opis dekodera w języku CUPL, który po skompilowaniu i wygenerowaniu wynikowego pliku programującego umożliwił zaprogramowanie matrycy pamięciowej US1. Uważni Czytelnicy zauważą na list. 1, że sygnał z linii adresowej A0 jest „przepuszczalny” przez US1 bez żadnych modyfikacji. Jest to pozostałość po próbach prowadzonych na egzemplarzu modelowym, bez większego znaczenia dla praktycznego działania karty. Dekoder jest tak

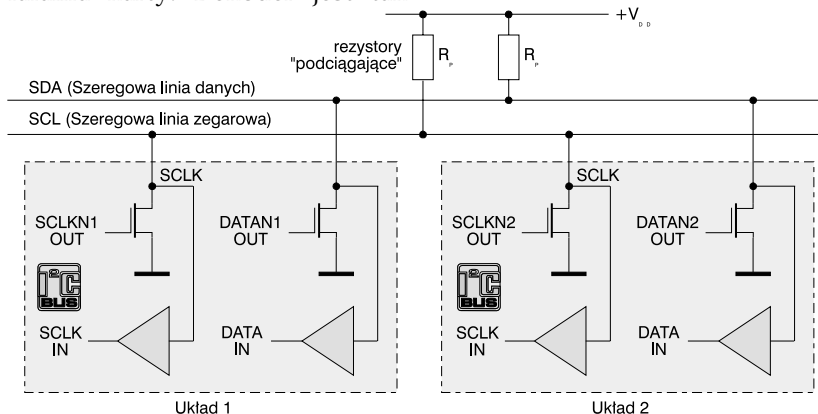
skonstruowany, że po pierwszym wpisie dowolnej danej pod adres wybrany przez użytkownika, układ PCF8584 jest automatycznie skonfigurowany do pracy w trybie 80xx. Rozwiązanie takie okazało się niezbędne, ponieważ wejściowy interfejs sterujący US2 można zaadaptować także do współpracy z procesorami firmy Motorola. O sposobie skonfigurowania decyduje pierwszy dostęp do układu po jego wyzerowaniu.

Selekcji adresu bazowego karty można dokonać na dwa sposoby: poprzez zmianę położenia jumpera na złączu JP1 lub poprzez przeprogramowanie układu ispGDS14. Na rys. 12 przedstawiona została taka modyfikacja układu z rys. 11, aby w miejsce złącza JP1 zainstalować układ US3.

Ponieważ układy ispGDS są stosunkowo mało znane wśród elektroników w naszym kraju, pokrótce przedstawimy ich strukturę i możliwości.



Rys. 10. Wykres przedstawiający zależność pomiędzy wartością  $R_p$  i pojemnością linii magistrali.



Rys. 9. Budowa portów I/O typowych układów I<sup>2</sup>C.

### WYKAZ ELEMENTÓW

#### Wersja podstawowa

#### Rezystory

- R1: 2,7kΩ
- R2, R3: 68Ω
- R4, R5: 5,1kΩ

#### Kondensatory

- C1, C2, C3, C4: 100nF
- C5, C6: 10μF/10V
- C7: 47μF/10V

#### Półprzewodniki

- US1: GAL20V8B - zaprogramowany AVT-265
- US2: PCF8584P
- D1, D2: 1N4148 lub podobne

#### Różne

- X1: 6MHz oscylator
- JP1: podwójna, pięciopozycyjna listwa gold-pin z jumperem
- G1: gniazdo Mini DIN6
- Sw1: przycisk chwilowy
- Śledz mocujący kartę

#### Opcje

- US3: ispGDS14-7J zamiast JP1 (z podstawką PLCC20)
- X1: DS1075M-060ES lub -066ES zaprogramowany 6MHz (zamiast standardowego generatora 6MHz)
- G2: 8-stykowe gniazdo telefoniczne (stosować tylko w przypadku zastąpieniu JP1 układem US3)

### Listing 1.

```

NAME avt265;
REV 1.2;
DESIGNER Piotr Zbysinski;
COMPANY B T C;
DEVICE g20v8;

/* ***** */
/* * */
/* * Dekoder adresowy do karty I2C * */
/* * */
/* ***** */

/* INPUTS */
pin [1..10] = [A0..A9];
pin 11 = RES_IN;
pin 13 = IOWR;
pin 14 = WR_ENA;
pin 23 = AEN;

/* OUTPUTS */
pin [17..21] = ![SEL4..0];
pin 15 = !RES_OUT;
pin 16 = !WR_OUT;
pin 22 = A0_OUT;

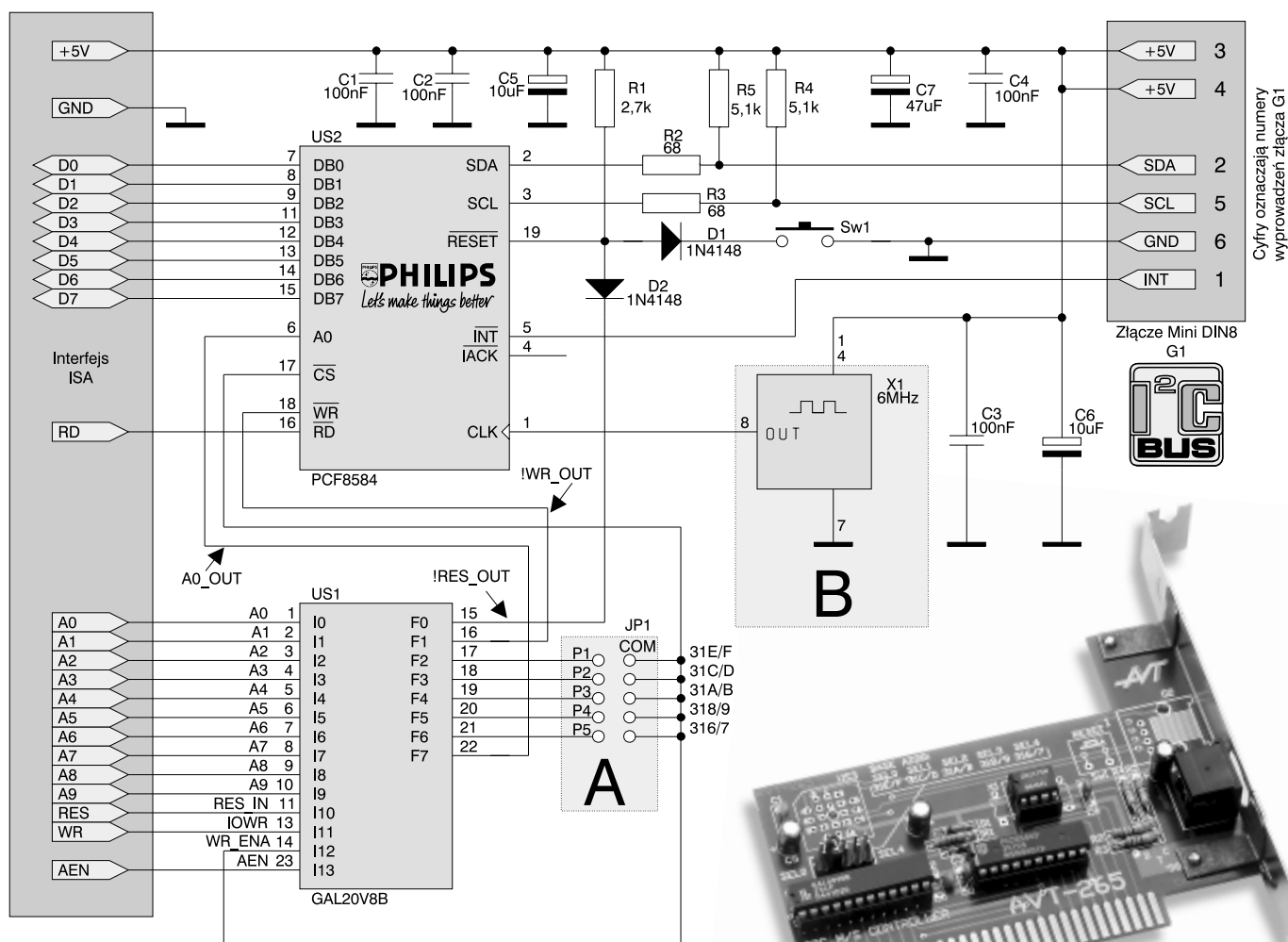
/* DECLARATIONS AND INTERMEDIATE VARIABLE */
/* DEFINITIONS */
field ADDRESS = [A0..A9];

/*LOGIC EQUATIONS*/
SEL0 = ADDRESS:'h'31e # ADRES:'h'31f;
SEL1 = ADDRESS:'h'31c # ADRES:'h'31d;
SEL2 = ADDRESS:'h'31a # ADRES:'h'31b;
SEL3 = ADDRESS:'h'318 # ADRES:'h'319;
SEL4 = ADDRESS:'h'316 # ADRES:'h'317;

RES_OUT = RES_IN;

A0_OUT = A0;

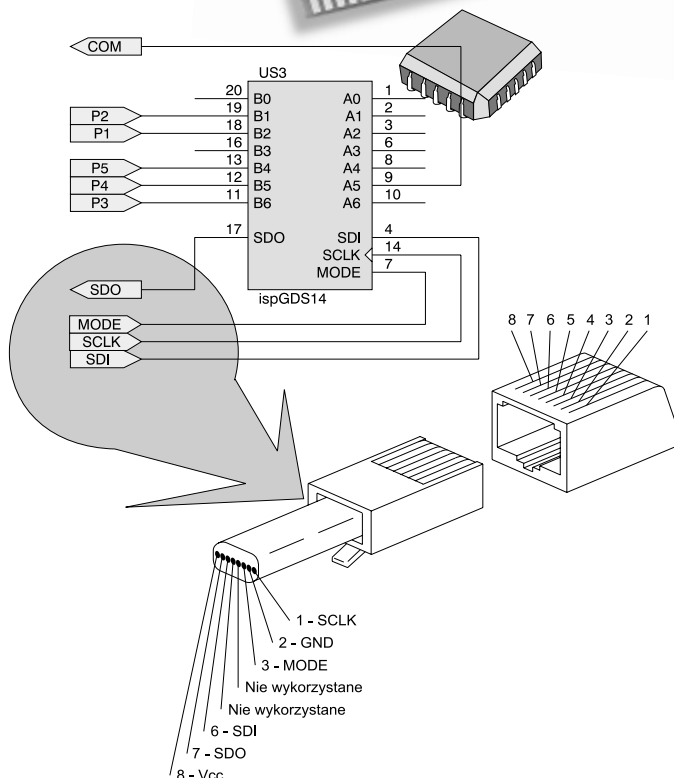
WR_OUT = !WR_ENA & !IOWR;
    
```



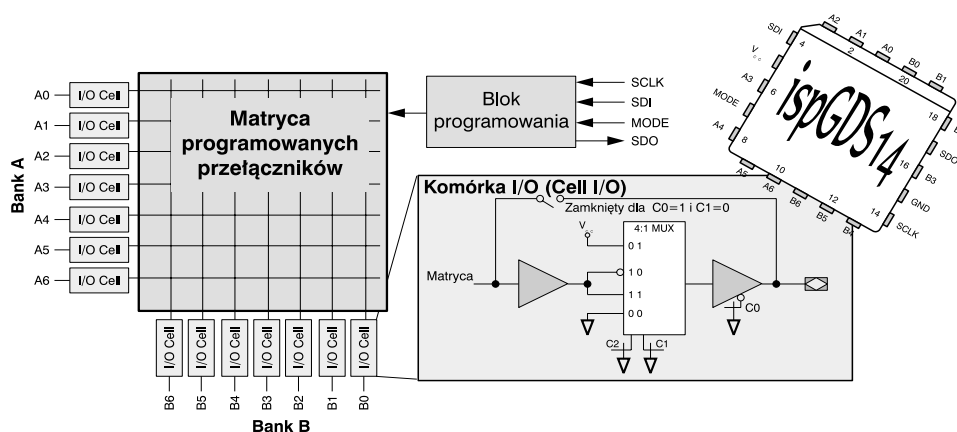
Rys. 11. Schemat elektryczny interfejsu.

Seria układów GDS (ang. Generic Data Switch) jest autorskim opracowaniem firmy Lattice. Podobnie do innych układów programowalnych *isp* są one reprogramowalne w systemie, poprzez prosty interfejs szeregowy.

Wewnętrzna struktura układów *ispGDS* przypomina matrycę przełączników, które można w dowolny sposób programować (włączone/wyłączone). Układ *ispGDS14* ma dwa porty I/O (Bank A i Bank B, rys. 13), których wyprowadzenia można ze sobą łączyć (programować) w dowolnym kierunku, możliwe jest negowanie łączonych sygnałów lub ustalanie na wybranym wyjściu poziomu logicznego „na stałe” (rys. 14). Tak więc układ *ispGDS14* doskonale nadaje się do zastąpienia kilku standardowych przełączników wykorzystywanych do konfiguracji lub programowania urządzenia, przy czym zmiana „położenia” przełącznika wymaga każdorazowej zmiany pliku programującego wpisywanego do matrycy pamięciowej układu.



Rys. 12. Schemat elektryczny przedstawiający sposób zastąpienia JP1 układem *ispGDS14* (moduł A na schemacie z rys. 11).



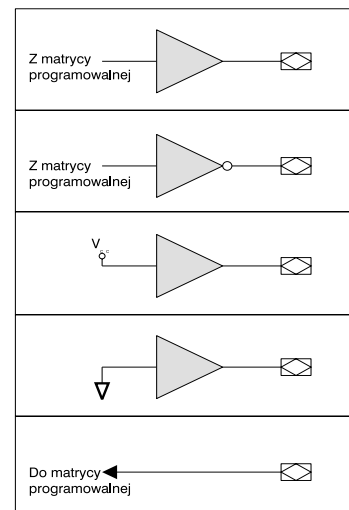
Rys. 13. Budowa wewnętrzna układu ispGDS14.

Układ US2 musi być taktowany prostokątnym sygnałem zegarowym, który jest generowany przez oscylator X1. W egzemplarzu modelowym zastosowano oscylator kwarcowy o częstotliwości 6MHz, przeprowadzono także próbę zastąpienia tego oscylatora przez scalony, programowany generator DS1075 firmy Dallas. Sposób jego wykorzystania przedstawimy w kolejnej części artykułu.

Rezystor R1 „podciąga” do plusa zasilania wejście zerujące US2. Diody D1 i D2 tworzą funkcyjny

logiczny AND, dzięki któremu układ US2 jest każdorazowo zerowany po włączeniu komputera i może być zerowany także ręcznie przy pomocy przycisku Sw1. Rezystory R4 i R5 „podciągają” linie SCL i SDA do plusa zasilania, a R2 i R3 zapobiegają możliwości uszkodzenia obwodów wejściowych US2.

Wszystkie sygnały I<sup>2</sup>C oraz linie zasilające wyprowadzone zostały na złącze G1. Zastosowano typowe 6-stykowe gniazdo Mini-DIN, zalecane przez firmę Philips



Rys. 14. Możliwość skonfigurowania układu ispGDS14.

do stosowania w magistralach ACCESS.bus (pochodna I<sup>2</sup>C). Umożliwia ono proste dołączenie dowolnego urządzenia zewnętrznego, bez konieczności ingerencji we wnętrze komputera.

**Piotr Zbysiński, AVT**

*W artykule wykorzystano materiały firmy Philips, opublikowane w katalogu „I<sup>2</sup>C Peripherals” z roku 1996.*