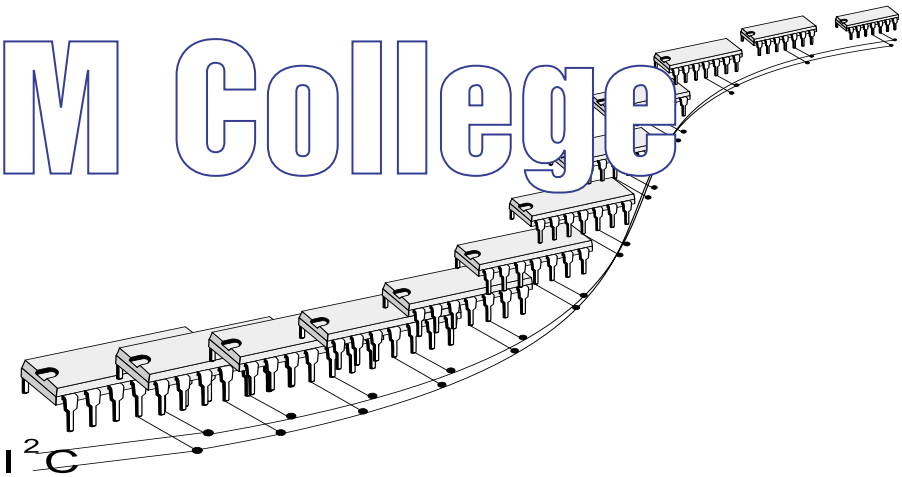
BASCOM College



Wykład 6

Magistrala I²C

Serdecznie witam Studentów i Sympatyków BASCOM College na kolejnym wykładzie. Będzie to lekcja w pewnym stopniu wyjątkowa, ponieważ omawiany na niej będzie szczególnie ważny i dość trudny temat. Po raz pierwszy będziemy musieli zerwać z przestrzeganą dotąd zasadą niewgłębiania się w teorię i traktowania elementów systemu mikroprocesorowego po trosze jako "czarnych skrzynek", które działają, ale nieważne jak. Tematem wykładu będzie jeden z najważniejszych składników wielu systemów mikroprocesorowych, w szczególności pracujących w urządzeniach elektronicznych powszechnego użytku. Składnik ten jest dla systemu mikroprocesorowego tym samym, czym dla żywego organizmu jest układ nerwowy: pozwala on przenosić informacje od i z układów peryferyjnych. Oczywiście, domyśliliście się już, że mam na myśli magistralę I²C.

Rozbudowywanie systemów mikroprocesorowych poprzez dołączanie do nich dodatkowych układów peryferyjnych zawsze było sprawą kłopotliwą, a to głównie ze względu na ograniczoną liczbę wyprowadzeń mikroprocesorów. "Typowy" procesor posiada ich co prawda najczęściej 32 (cztery ośmiobitowe porty), ale ostatnio pojawia się coraz więcej procesorów z ograniczoną liczbą wyprowadzeń, w dwudziesto- albo nawet ośmiopinowych obudowach. Procesory te, powszechnie dostępne i relatywnie tanie, zyskały sobie wielkie uznanie u amatorów i profesjonalistów, ale ich stosowanie jeszcze zwiększa problemy z ograniczoną liczbą wyprowadzeń.

Z drugiej strony, projektant systemu mikroprocesorowego, w założeniu o dużym stopniu uniwersalności, chciałby nieraz zapewnić sobie możliwości rozbudowy układu w miarę pojawiających się nowych problemów i wymagań użytkowników. Wielu konstruktorów marzyło o systemie procesorowym składającym się z jednostki centralnej i dołączanych do niej za pomocą uniwersalnego łącza układów peryferyjnych, których ani liczba ani rodzaj nie musiałby być z góry definiowany. W odpowiedzi na te słuszne żądania firma Philips opracowała nowy standard w systemach mikroprocesorowych: magistralę I²C.

Zasady działania magistrali I²C nie są bynajmniej proste, a szczegółowy jej opis wykroczyłby z pewnością poza ramy tego wykładu. Omówmy więc te sprawy dość ogólnikowo, zainteresowanych odsyłając do specjalistycznej literatury. Pamiętajmy, że o zachowanie zgodności naszego programu z przyjętymi standardami dba "mądry" kompilator, a transmisja danych w magistrali I²C nie jest tutaj wyjątkiem.

Do czego właściwie ta magistrala będzie potrzebna? Przecież sam procesor "może wszystko", a zdolny konstruktor powinien tak zaprojektować układ, aby zawsze wystarczyło wyprowadzeń do obsługi potrzebnych funkcji. W ostateczności, możemy przecież zawsze zastosować "większy" procesor, niewiele droższy od naszego '2051. Niestety, nie zawsze będzie to możliwe. Bywa, i to wcale nie w bardzo skomplikowanych układach, że nawet 32 aktywne wyprowadzenia dużej „pięćdziesiątki jedynki"' nie wystarczają. W systemach automatyki zachodzi nieraz potrzeba sterowania tak wieloma układami, że musimy sięgnąć po ekspandery, zwiększające liczbę aktywnych wyprowadzeń procesora. A te ekspandery, jak chociażby znany Wam PCF8574, obsługiwane są właśnie "**i kwadratem**".

Jest jeszcze jeden powód, dla którego powinniśmy zgłębić tajniki magistrali I²C. Istnieje ogromna ilość bardzo fajnych układów, z których zbudować możemy wiele interesujących i, co bardzo ważne, prostych urządzeń, które obsługiwane są wyłącznie przez magistralę I²C. Mam tu na myśli przede wszystkim układy przeznaczone do pracy w sprzęcie RTV i audio. Zbudowanie np. dekodera Surround bez stosowania procesora i transmisji I²C jest w obecnej chwili praktycznie niemożliwe. Sercem większości współczesnych systemów audio jest przecież procesor, którego zadaniem jest sterowanie niekiedy bardzo skomplikowanymi funkcjami systemu. Tak, tak, do tego właśnie doszliśmy: dziesiątki oporniczków, kondensatorków i wzmacniaczy operacyjnych zostały już dawno zastąpione procesorem i wyspecjalizowanymi układami scalonymi. Tylko że Wy do tej pory nie mieliście okazji się z nimi zapoznać. Po raz drugi uchylę rąbka tajemnicy i powiem Wam, że w ostatnim stadium testowania znajduje się, przeznaczony do publikacji w EdW, wzmacniacz Surround SRS zbudowany na dwóch układach scalonych (nie liczę tu końcówek mocy, bo to już inna bajeczka): procesorze '2051 i wyspecjalizowanym układzie TDA7431. No i co, przekonałem Was? To bierzemy się do nauki!

Zainstalowanie magistrali I²C w systemie mikroprocesorowym, do którego program będzie pisany w pakiecie BASCOM, jest wyjątkowo proste. Możemy uczynić to na dwa sposoby. Sposobem najprostszym jest zadeklarowanie na początku programu, które wyprowadzenie procesora ma obsługiwać linię SDA, a które SCL. Wybór jest całkowicie dowolny, a wiec najczęściej będziemy kierować się chęcią maksymalnego uproszczenia projektowania płytki obwodu drukowanego. Polecenia konfigurujące magistralę I²C mają postać:

Config SDA = pin
Config SCL = pin
np.
Config Scl = P3.0
Config Sda = P3.1

Drugim sposobem zainstalowania magistrali I²C w systemie jest wybranie z menu OPTIONS opcji COMPILER\I²C i wpisanie definicji właściwych pinów procesora w odpowiednich okienkach (**rysunek 1**). **Uwaga: zdefiniowanie wyprowadzeń magistrali I²C w programie ma priorytet przed określeniem ich w opcjach kompilatora. Jeżeli podane wyprowadzenia różnią się pomiędzy sobą, to kompilator zignoruje dane wpisane w okienku konfiguracyjnym**.

Rys. 1

Ogólna charakterystyka magistrali I²C

W aplikacjach 8-bitowych, np. takich, gdzie stosowany jest mikrokontroler jednoukładowy, można przyjąć pewne założenia projektowe:

- System składa się z co najmniej jednego mikrokontrolera, pamięci i układów wejścia/wyjścia.

- Koszt połączeń różnych elementów systemu musi być jak najmniejszy. System taki spełnia najczęściej funkcje sterujące i nie jest wymagana duża szybkość przesyłania danych.

- Efektywność całego systemu zależy od wybranych układów i struktury magistrali łączącej poszczególne jego elementy.

Aby spełnić powyższe założenia, wymagana jest szeregowa struktura magistrali. Mimo iż magistrale szeregowe nie mają zdolności przepływowych magistral równoległych, wymagają mniej przewodów i wyprowadzeń. Jednakże magistrala nie jest jedynie przewodem połączeniowym, zawiera także wszystkie formaty i procedury komunikacji wewnątrz systemu.

Urządzenia komunikujące się między sobą po magistrali szeregowej muszą mieć pewnego rodzaju protokół chroniący przed wszelkimi możliwościami kolizji, utraty danych i zablokowaniem wymiany informacji. Urządzenia szybkie muszą umieć wymieniać dane z urządzeniami wolnymi. System nie może być zależny od urządzeń do niego dołączonych - peryferyjnych, ponieważ w przeciwnym przypadku nie byłoby możliwe modyfikowanie i ulepszanie takiego systemu. Procedura komunikacji musi być w stanie zdecydować, które urządzenie będzie sterować magistralą i kiedy. W przypadku dołączenia do szyny urządzeń o rożnych szybkościach zegara, musi być określone źródło sygnału zegarowego magistrali. Wszystkie te kryteria zostały uwzględnione w specyfikacji magistrali I²C.

Układy scalone dołączane do magistrali I²C mogą być wykonane w dowolnej technologii (NMOS, CMOS, bipolarna). Dwa przewody, danych szeregowych SDA (ang. Serial Data) i zegara szeregowego SCL (ang. Serial Clock), przenoszą informację pomiędzy urządzeniami dołączonymi do magistrali. Każde urządzenie jest rozpoznawane przez unikalny adres, niezależnie czy jest to mikrokontroler, sterownik wyświetlacza LCD, pamięć lub interfejs klawiatury - i może pracować jako nadawca lub odbiorca, zależnie od realizowanej funkcji. Oczywiście, np. sterownik wyświetlacza LCD może najczęściej być tylko odbiorcą, podczas gdy pamięć może być nadawcą i odbiorcą. Dodatkowo, w czasie przeprowadzania transmisji danych, urządzenie może być urządzeniem nadzorującym (ang. Master) lub urządzeniem podporządkowanym (ang. Slave) (patrz Tabela l). Urządzeniem nadzorującym jest to urządzenie, które inicjuje transfer danych i generuje sygnał zegarowy do jego wykonania. W tym czasie dowolne urządzenie zaadresowane jest urządzeniem podporządkowanym.

Jeśli dwa lub więcej urządzeń nadzorujących próbuje wystawić dane na szynę, pierwsze, które wystawia stan wysoki (podczas gdy inne wystawia stan niski), traci dostęp do magistrali. W czasie arbitrażu sygnały zegarowe są synchronizowane kombinacją zegarów generowanych przez urządzenia nadzorcze. Synchronizację uzyskuje się przez połączenie "wired-AND" sygnałów zegarowych poszczególnych urządzeń do linii SCL.

Za generację sygnałów zegarowych zawsze odpowiada urządzenie nadzorujące; każde takie urządzenie w czasie transmisji danych wytwarza swój własny sygnał zegarowy. Częstotliwość sygnału zegarowego nie jest w żaden sposób ograniczona “od dołu” (ma to szczególne znaczenie podczas przeprowadzania emulacji sprzętowej). Magistralowe sygnały zegarowe wytwarzane przez urządzenie nadzorcze mogą być zmieniane tylko w przypadku kiedy są rozciągane przez powolne urządzenia podporządkowane (ang. Slow-Slave Device), utrzymujące stan niski na linii zegara, lub przez inne urządzenie nadzorcze w czasie arbitrażu.

Różnorodność technologii, w jakich mogą zostać wykonane urządzenia I²C (CMOS, NMOS, bipolarna) powoduje, że poziomy zera logicznego (stan niski) i logicznej jedynki (stan wysoki) nie zostały ustalone i są związane z poziomem napięcia zasilającego. Jest to fakt bardzo wygodny dla projektanta systemu, ponieważ pozostawia szerokie pole do wyboru typu układów pracujących na magistrali I²C i poziomu zasilającego je napięcia.

Definicje terminologii magistrali I²C	
Nadawca (ang. Transmitter)	Urządzenie, które wysyła dane na magistralę
Odbiorca (ang. Receiver)	Urządzenie, które odczytuje dane z magistrali
Urządzenie nadzorujące (ang. Master)	Urządzenie, które inicjuje przesyłanie danych, generuje sygnał zegarowy i kończy transfer
Urządzenie podporządkowane (ang. Slave)	Urządzenie zaadresowane przez urządzenie nadzorujące
Wielodostęp (ang. Multi-Master)	W tym samym czasie więcej niż jedno urządzenie nadzorujące może próbować sterować magistralą bez zakłócania wiadomości
Arbitraż (ang. Arbitration)	Procedura zapewniająca, że w przypadku jeśli jednocześnie dwa (lub więcej) urządzenia nadzorujące próbują sterować magistralą, tylko jedno uzyska nad nią kontrolę, a transmitowana wiadomość nie zostanie zakłócona
Synchronizacja (ang. Synchronization)	Procedura synchronizująca sygnały zegarowe dwóch lub więcej urządzeń

Tabela 1

Do magistrali I²C może być dołączonych wiele urządzeń posiadających możliwość sterowania tą szyną (ang. Multi-Master Bus). Zazwyczaj urządzeniami nadzorującymi są mikrokontrolery, ale możliwe jest także zastosowanie w tej roli komputera PC.

Możliwość dołączenia więcej niż jednego mikrokontrolera do magistrali I²C oznacza, że więcej niż jeden nadzorca może próbować w tym samym czasie rozpocząć transmisję danych. By uniknąć chaosu jaki może powstać w takich przypadkach opracowano procedurę arbitrażu. Procedura ta wykorzystuje połączenie typu "wired-AND" (tzw. iloczyn montażowy) wszystkich interfejsów linii magistrali I²C.

Dane na linii SDA muszą być stabilne w czasie trwania stanu wysokiego na linii SCL. Stan wysoki lub niski na linii danych może się zmieniać tylko w czasie, gdy na linii zegara występuje niski stan logiczny.

Rozpoczęcie i zakończenie transmisji

W czasie działania magistrali I²C pojawiają się wyróżnione sekwencje stanów logicznych na liniach SDA i SCL, które definiują rozpoczęcie i zakończenie transmisji.

Przejście z wysokiego do niskiego stanu logicznego linii SDA, w czasie gdy SCL jest w stanie wysokim oznacza rozpoczęcie transmisji. W języku MCS BASIC inicjalizacja magistrali I²C odbywa się po wydaniu polecenia:

I2CSTART

Przejście z niskiego do wysokiego stanu logicznego linii SDA, w czasie gdy SCL jest w stanie wysokim, oznacza koniec transmisji. W języku używanym w naszym BASCOM-ie zakończenie transmisji następuje po wydaniu polecenia:

I2CSTOP

Sekwencje rozpoczęcia (START) i zakończenia (STOP) transmisji generowane są zawsze przez urządzenie nadzorujące. Magistrala jest postrzegana jako zajęta po wystąpieniu sekwencji START. Po pewnym czasie od wystąpienia sekwencji STOP, szyna jest ponownie wolna.

Wykrywanie sekwencji START i STOP przez urządzenia dołączone do magistrali I²C jest łatwe, o ile zawierają one konieczne rozwiązania sprzętowe. Jednakże mikrokontrolery nie posiadające takich interfejsów muszą próbkować linię SDA przynajmniej dwa razy w czasie każdego cyklu zegara w celu wykrycia zmiany stanu. Niestety, używane przez nas procesory '2051 wykorzystywać muszą tę drugą metodę. Na szczęście, w konstrukcjach amatorskich stosowanie procesora jako urządzenia podporządkowanego magistrali I²C zdarza się wyjątkowo rzadko.

Każdy bajt wysłany na linię SDA magistrali musi zawierać 8 bitów, nawet gdyby to miały być bity nieznaczące. Liczba bajtów, które mogą być przesyłane w czasie jednego transferu jest nieograniczona. Po każdym bajcie musi wystąpić bit potwierdzenia (ang. Acknowledge ACK). Jako pierwszy przesyłany jest bit najbardziej znaczący (ang. Most Significant Bit - MSB). Jeśli odbiorca nie może odebrać kolejnego pełnego bajtu danych do czasu wykonania innej funkcji, np. obsługi wewnętrznego przerwania, może utrzymać stan niski na linii zegara SCL zmuszając w ten sposób nadawcę do przejścia w tryb oczekiwania. Przesyłanie danych jest kontynuowane kiedy odbiorca jest gotowy do odebrania następnego bajtu i zwolni linię zegara SCL. Dodatkowy impuls zegara związany z przesyłaniem potwierdzenia (cykl potwierdzenia) generowany jest przez nadzorcę. W czasie trwania tego impulsu, nadawca zwalnia linię SDA (stan wysoki). W czasie trwania cyklu potwierdzenia odbiorca musi wystawić na linii SDA niski stan logiczny, tak by pozostawał on stabilny podczas trwania stanu wysokiego na linii zegara SCL.

Każde urządzenie pracujące na magistrali I²C musi posiadać swój własny, niepowtarzalny identyfikujący je adres. **Zainstalowanie na magistrali I²C dwóch układów o identycznym adresie jest absolutnie niedopuszczalne i, jak wykazuje praktyka jest najczęstszą przyczyną kłopotów podczas uruchamiania systemów mikroprocesorowych.**

Maksymalna liczba układów, jakie mogą zostać jednocześnie zainstalowane na standardowej magistrali I²C wynosi teoretycznie 255. Celowo napisałem "teoretycznie", ponieważ trudno przypuścić, aby komuś był potrzebny aż tak rozbudowany system mikroprocesorowy. Ponadto teoretyczna liczba układów ograniczana jest przez najrozmaitsze czynniki, między innymi przez dwoistość adresów niektórych urządzeń.

Większość układów I²C posiadających możliwość zarówno przyjmowania, jak i wysyłania danych posiada nawet dwa adresy: jeden do zapisywania w nich informacji, a drugi do ich odczytywania. Przykładowo, dobrze nam już znana kostka PCF8574A posiada adres bazowy do zapisu 162, a do odczytu 163.

Wiele układów dołączanych do magistrali I²C posiada także możliwość sprzętowej zmiany adresu, najczęściej za pomocą zwierania lub rozwierania z masą odpowiednich wyprowadzeń adresowych. Dla przykładu, wielokrotnie już wspominany ekspander PCF8574A posiada trzy sprzętowe wejścia adresowe, co umożliwia symultaniczną pracę ośmiu takich układów na jednej magistrali I²C.

W tabeli 2, na przykładzie układu PCF8574A, została pokazana składnia słowa adresowego, stosowanego do wywoływania urządzeń pracujących na I²C, a w tabeli 3 adresowanie jednego urządzenia, posiadającego możliwość sprzętowej zmiany adresu.

Tabela 2

Budowa słowa adresowego układu PCF8574A							
MSB							LSB
0	1	1	1	X	X	X	X
Wartość stała, charakterystyczna dla grupy układów, 0100 dla PCF8574				Pod-adres konfigurowany sprzętowo			0 - zapis, 1- odczyt

Adresowanie układu PCF8574A				
A2	A1	A0	Adres do zapisu	Adres do odczytu
0	0	0	112	113
0	0	1	114	115
0	1	0	116	117
0	1	1	118	119
1	0	0	120	121
1	0	1	122	123
1	1	0	124	125
1	1	1	126	127

Tabela 3

W języku MCS BASIC wysyłanie bajtu na magistralę I²C realizowane jest za pomocą polecenia:

I2CWBYTE wartość bajtu, [ACK (9), NACK (8)]

które oczywiście musi być poprzedzone inicjalizacją magistrali (I²CSTART) oraz podaniem adresu urządzenia podporządkowanego. Dodanie dyrektywy **ACK** lub **"9"** sygnalizuje, że mają być jeszcze wysłane następne bajty i że wymagane jest potwierdzenie odbioru bieżącego bajtu. Dyrektywa **NACK** lub **"8"** zwalnia urządzenie podporządkowane z konieczności potwierdzania odebrania transmisji.

Cały pakiet poleceń potrzebnych do wysłania bajtu na szynę danych magistrali I2C powinien np. wyglądać następująco:

I2cstart
I2cwbyte [adres urządzenia podporządkowanego do zapisu], Ack
I2cwbyte [bajt do wysłania]
..... (wysyłanie kolejnych bajtów)
I2cwbyte [ostatni wysłany bajt], Nack
I2cstop

Poleceniem komplementarnym do I²wbyte jest I²crbyte, służące do odczytywania danych z układu podporządkowanego. Jego składnia jest praktycznie identyczna ze składnią polecenia I²cwbyte:

I2CRBYTE wartość bajtu, [ACK (9), NACK (8)]

a przykładowy program pozwalający na odczytanie z układu podporządkowanego kilku bajtów powinien wyglądać następująco:

I2cstart
I2crbyte [adres urządzenia podporządkowanego do odczytu], Ack
..... (wysyłanie kolejnych bajtów)
I2crbyte [ostatni odebrany bajt], Nack
I2cstop

Nieco większe możliwości daje polecenie:

I2CSEND adres urządzenia podporządkowanego, wartość bajtu, ilość bajtów

za pomocą którego możemy wysłać na magistralę I²C dowolną liczbę bajtów, np. umieszczonych przedtem w tablicy.

Na przykład:

```
Dim bajt (10) As Byte   'deklaracja tablicy zawierającej 10 zmiennych
Slave = &H40            'podanie adresu urządzenia odporządkowanego
                                        (konkretnie PCF8574)
For a = 1 to 10         'dziesięciokrotnie
   bajt(a) = a          'nadaj zmiennej "bajt" przykładowe
                                        wartości (od 1 do 10)
Next
I2CSEND slave, bajt(), 10   ' wyślij do odbiorcy 10 bajtów
End
```

Ciąg dalszy na stronie 26

Ciąg dalszy ze strony ????

Warto zauważyć, że polecenie I2CSEND nie wymaga wstępnego inicjalizowania magistrali, które wykonywane jest samoczynnie. Niestety, większa uniwersalność tego polecenia okupiona jest pewnym zwiększeniem długości kodu wynikowego.

Poleceniem komplementarnym do I2CSEND jest **I2CRECEIVE**. Składnia tego polecenia i jego możliwości są bardzo podobne do I2CSEND. W najprostszej postaci używamy tego polecenia do odczytywania jednej wartości z urządzenia podporządkowanego:

I2CRECEIVE [adres, wartość]

Jednak składnia polecenia I2CRECEIVE może być znacznie bardziej rozbudowana, a to samo polecenie może służyć zarówno do odbierania danych z magistrali I2C, jak i do ich wysyłania:

I2CRECEIVE [adres, wartość ,ilość bajtów do wysłania, ilość bajtów do odebrania]

Na przykład:

```
Dim Wartosc(10) As Byte
Wartosc(1) = 1
Wartosc(2) = 3
I2creceive adres urządzenia, Wartosc(), 2, 1
                        'wysłanie na magistralę
      I2C dwóch bajtów i odebranie 'jednego bajtu
Print Wartosc(1)        'wydruk odebranej wartości
```

Na tym możemy zakończyć teoretyczne rozważania na temat magistrali I2C i sposobu jej obsługiwania z poziomu języka MCS BASIC. Zapraszam Was teraz do wykonania kilku ćwiczeń, które dadzą nam rzecz najważniejszą: praktyczną wiedzę o sposobach wykorzystywania magistrali I2C.

Zbigniew Raabe,
e-mail: **zbigniew.raabe@edw.com.pl**
Konsultacje: **Sławomir Surowiński**,
e-mail: **slawomir.surowinski@ep.com.pl**