

# Interfejs I<sup>2</sup>C slave w mikrokontrolerach AVR

Zaprojektowany przez Philipsa interfejs I<sup>2</sup>C pozwala prowadzić komunikację pomiędzy inteligentnymi modułami, a nawet pojedynczymi układami systemów cyfrowych, przy wykorzystaniu bardzo prostej, dwuprzewodowej magistrali szeregowej. Interfejs ten stał się jednym z najbardziej popularnych standardów stosowanych przez wielu światowych producentów, choć z przyczyn licencyjnych nie zawsze występuje jako I<sup>2</sup>C.

**Rekomendacje:** przykłady przedstawione w artykule należy traktować jako wskazówkę i inspirację przydatną podczas realizacji własnych projektów.

Sposób dołączania układów do magistrali I<sup>2</sup>C pokazano na rys. 1. Układy mogą pracować w trybach master (nadrzędnym) lub slave (podrzędnym). Układ master steruje transmisją, slave tylko transmituje dane w odpowiedzi na rozkazy układu nadrzędnego. W jednej magistrali I<sup>2</sup>C może pracować kilka układów nadrzędnych i podrzędnych. Stosowane są dwa standardy interfejsu – pierwszy przewiduje prowadzenie transmisji z prędkościami do 100 kbit/s (tryb normalny), w drugim prędkość transmisji może dochodzić do 400 kbit/s (tryb szybki). Dopuszczalna długość połączeń oraz prędkość transmisji ograniczone są pojemnościami pasożytniczymi występującymi

w magistrali. Do transmisji używa się dwóch linii: SDA – linia danych i SCL – linia zegarowa. Każdy układ slave z magistralą I<sup>2</sup>C jest rozpoznawany przez unikatowy adres, niezależnie od tego, czy jest to mikrokontroler, pamięć, czy inny układ. Obydwie linie (SDA i SCL) są liniami dwukierunkowymi i muszą być podciągnięte do plusa zasilania przez rezystory zewnętrzne (rys. 1). Ich rezystancja wynosi zazwyczaj 4,7 kΩ. W stanie spoczynku (gdy dane nie są przesyłane) na obu liniach występuje wysoki poziom logiczny. Dane przesyłane magistralą I<sup>2</sup>C są 8-bitowe. Każda wymiana danych rozpoczyna się sekwencją startu, a kończy sekwencją stopu. Każde pra-

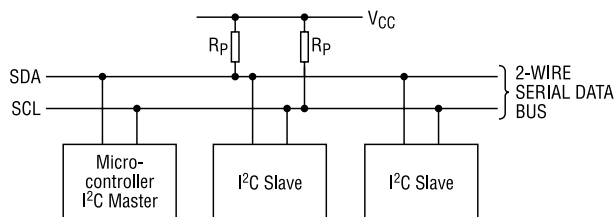
**Dodatkowe materiały na CD i FTP:**  
host: ep.com.pl, user: 12235, pass: 60u61csy  
• listingi do artykułu

widlowe przesłanie bajtu danych jest sygnalizowane sekwencją potwierdzenia (ACK).

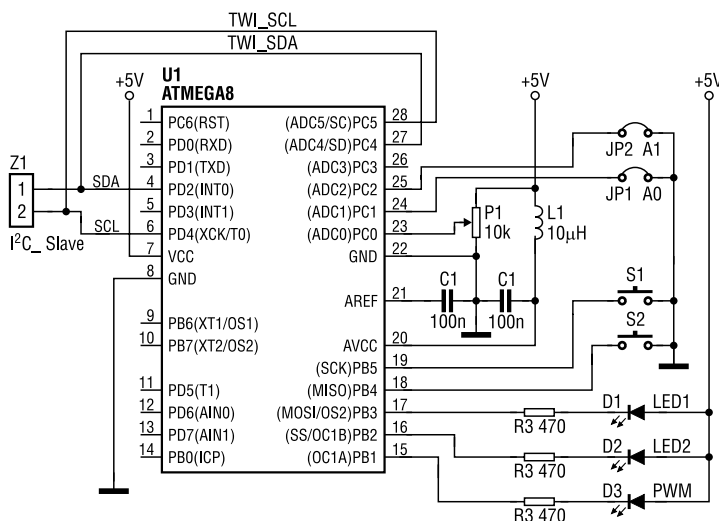
Urządzenia konstruowane wspólnie najczęściej budowane są jako zespół specjalizowanych bloków połączonych ze sobą w jeden system. Każdy z takich bloków realizuje przydzieloną mu funkcję, a wszystkim steruje najczęściej mikrokontroler. Aby urządzenie takie mogło działać prawidłowo, konieczne jest komunikowanie się między sobą poszczególnych bloków, czasami bezpośrednio, czasami poprzez nadzorujący mikrokontroler. Do prowadzenia takiej łączności można stosować wiele istniejących standardów. Jednym z najlepszych w tym zakresie jest I<sup>2</sup>C. W celu uniknięcia chaosu jeden z tych bloków pełni funkcję master, pozostałe bloki (każdy z nich może zawierać własny mikrokontroler) są podporządkowane.

Wiele specjalizowanych układów ma fabrycznie zaimplementowany interfejs I<sup>2</sup>C, pracujący najczęściej jako slave. Nie zawsze jednak układy takie nadają się do bezpośredniego zastosowania w danym urządzeniu. Jednym ze sposobów poradzenia sobie z tego typu problemami może być implementacja podrzędnego interfejsu I<sup>2</sup>C w mikrokontrolerze. Dzięki takiemu rozwiązaniu (głównie dzięki możliwości dość swobodnego zastosowania oprogramowania) można uzyskać dowolny algorytm działania układu podrzędnego.

Poniżej zostanie przedstawiona programowa oraz sprzętowa implementacja podrzędnego interfejsu I<sup>2</sup>C w zasobach mikrokontrolera AVR. Należy zwrócić uwagę na to, że niektóre mikrokontrolery AVR mają wbudowany interfejs I<sup>2</sup>C (nazywany przez firmę Atmel – TWI), mogący pracować jako układ nadrzędny lub podrzędny. Programowy interfejs I<sup>2</sup>C slave można wykorzystywać zwłaszcza w małych mikrokontrolerach niezawierających sprzętowego TWI. Zostaną przedstawione przykłady realizacji układu slave, z którego układ nadrzędny może odczytać stan dwóch linii wejściowych portu, wysłać do niego stan dwóch linii wyjściowych portu, odczytać wartość napięcia zmierzoną przez przetwornik A/C układu slave oraz wysłać do niego wartość wypełnienia generowanego przebiegu PWM również przez układ slave. Obie realizacje, programowa i sprzętowa, będą



Rys. 1. Łączenie układów do magistrali I<sup>2</sup>C



Rys. 2. Przykładowy schemat układu I<sup>2</sup>C Slave

identyczne pod względem funkcjonalnym. Przykłady interfejsu slave w mikrokontrolerze AVR zostaną przedstawione w oparciu o oprogramowanie BASCOM AVR. Pokazana zostanie również programowa i sprzętowa realizacja układu master z interfejsem I<sup>2</sup>C zaimplementowanego w zasoby mikrokontrolera. Będzie on odczytywał z układu slave stan linii wejściowych portu i zapisywał je do linii wyjściowych układu slave. Układ ten będzie również odczytany z przetwornika A/C wartość wysyłał do układu slave, traktując ją jako wartość wypełnienia przebiegu PWM generowanego przez układ slave. Odczytywane wartości z układu slave będą wyświetlane na wyświetlaczu LCD dołączonym do mastera. Komunikacja z układem slave będzie się odbywać z wykorzystaniem prostego protokołu, składającego się z adresu układu slave, adresu komendy i odczytywanej lub zapisywanej danej. Układ slave zrealizowany w oparciu o mikrokontroler umożliwia także wybór jego adresu, na który będzie odpowiadał układ.

### Programowa realizacja interfejsu I<sup>2</sup>C slave

Na rys. 2 został pokazany przykładowy schemat układu I<sup>2</sup>C slave. Porty, do których zostały dołączone przyciski S1 i S2, są skonfigurowane jako wejściowe. Ich stan może odczytywać układ master. Porty, do których zostały dołączone diody D1 i D2, pracują w trybie wyjściowym. Ich stanem również może sterować układ master. Wejściowa linia PC0 stanowi wejście 10-bitowego przetwornika A/C wbudowanego w mikrokontroler. Potencjometr P1 umożliwia zmianę napięcia mierzonego przez przetwornik A/C. Elementy L1, C1 i C2 są wymagane do zasilania przetwornika A/C i związanego z nim napięcia referencyjnego, które zostało ustalone na wartość 5 V. Wyjście sterujące diodą D3 pracuje w trybie generatora PWM. Jasność świecenia diody D3 będzie zależała od wypełnienia przebiegu PWM. Zworniki JP1 i JP2 umożliwiają wybór adresu układu slave. W przypadku programowej realizacji interfejsu I<sup>2</sup>C, linia SCL jest dołączona do portu PD4 mikrokontrolera, a linia SDA do PD2. Taki wybór linii mikrokontrolera związany jest z budową biblioteki *i2cslave.lib* znajdującej się w pakiecie Bascom AVR. Komu-

nikacja z układem slave (odczyt i zapis danych) może się odbywać za pomocą komend. W przykładzie będą potrzebne 2 komendy, które będą służyć do zapisu i odczytu stanu portów mikrokontrolera oraz do odczytu wartości z przetwornika A/C i zapisu wartości do generatora PWM. Przykładowe komendy dla opisywanego układu slave przedstawiono w tab. 1. Specyfikacja interfejsu I<sup>2</sup>C umożliwia wykorzystanie jednej komendy do zapisu i odczytu danych do różnych rejestrów. Rodzaj operacji (zapis lub odczyt) jest rozróżniany w adresie układu slave. W przypadku odczytu, adres układu slave jest o jeden większy niż podczas zapisu danych do układu. Przykładowo: jeśli adres zapisu układu jest równy 64, to przy odczycie będzie wykorzystywany adres 65. Oczywiście w przypadku interfejsu I<sup>2</sup>C należy wcześniej zapisać do układu adres komendy, która będzie informowała układ Slave, którą wartość rejestru ma wysłać do układu master. Na list. 1 (wszystkie listingi zamieszczono na CD-EP12/2009) pokazano przykład programowej realizacji interfejsu I<sup>2</sup>C slave. Do poprawnej pracy programowego interfejsu I<sup>2</sup>C slave wymagana jest biblioteka *i2cslave.lib*. Do konfiguracji programowego interfejsu I<sup>2</sup>C służy komenda *CONFIG I2CSLAVE*. Składnia tej komendy jest następująca:

**CONFIG I2CSLAVE = address , INT = interrupt , TIMER = tnr**  
gdzie:

ADDRESS – adres układu slave, musi być wartością parzystą (nieparzyste wartości adresu służą do odczytu danych z układu slave),

INT – domyślny parametr, który wskazuje na numer wykorzystywanego przerwania (domyślnie wykorzystywane jest zewnętrzne przerwanie INT0),

TIMER – domyślny parametr, który wskazuje na numer wykorzystywanego timera (domyślnie wykorzystywany jest Timer0).

Pomimo że nazwa wykorzystywanego przerwania jak i timera może być dodatkowo podana, to aby skorzystać z innego przerwania lub timera, należy przekonstruować zawartość biblioteki *i2cslave.lib*. Użycie linii PD.4 (T0), która jest wejściem Timera0 oraz linii PD.2 (INT0), która jest wejściem zewnętrznego przerwania jako linii interfejsu I<sup>2</sup>C slave, wymagane jest przez bibliotekę *i2cslave*. Timer oraz przerwanie INT0 są wykorzystywane do realizacji programowego interfejsu I<sup>2</sup>C Slave. Wynika z tego wniosek, że programowy interfejs I<sup>2</sup>C slave będzie działał tylko na tych mikrokontrolerach, które mają przynajmniej wejścia T0 i INT0. W programie z list. 1 w instrukcji *config i2cslave* został podany tylko adres, jaki ma mieć układ podrzędny. Ustalono go na wartość parzystą, równą 64. Kompilator automatycznie generuje bajt *\_i2c\_slave\_address*, w którym przechowywany jest adres układu sla-

ve. Zmieniając zawartość tego bajtu, można zmienić adres układu slave. Zostało to wykorzystane w przykładowym programie. Adres układu slave został uzależniony od stanu linii wejściowych PC1 i PC2 mikrokontrolera. Zmianę adresu układu slave zrealizowano w następujący sposób:

```
Adres_sl = Pinc And &B00000110
, odczyt stanu linii PC1 i PC2
określających adres Slave
_i2c_slave_address = 64 + Adres_sl
, zapis adresu Slave
```

W tab. 2 pokazano możliwe do wyboru wartości adresów w zależności od stanów na liniach PC1 i PC2 mikrokontrolera. Generowana jest także zmienna *\_i2c\_slave\_address\_received*, w której jest przechowywany odebrany od układu master adres. Kod biblioteki I<sup>2</sup>C slave wywołuje dwa podprogramy, których zawartość należy samemu przygotować i od której będzie zależało działanie układu slave. Jeśli układ Master żąda odczytania bajtu, wtedy wywoływany jest podprogram oznaczony etykietą *I2c\_master\_needs\_data*. Wysyłane dane do mastera należy umieszczać w zmiennej *\_a1*, która jest odzwierciedleniem rejestru R16 mikrokontrolera. W tym podprogramie umieszczono kod, który umożliwia w zależności od wysłanej przez master komendy, odczytanie wartości z przetwornika A/C oraz stanu dwóch linii portu PB. Jeśli układ master przesyła dane, wtedy wywoływany jest podprogram oznaczony etykietą *I2c\_master\_has\_data*. Przesyłane przez master dane są zapisywane w zmiennej *\_a1*. W tym podprogramie, w ramach przykładu, odczytywane są dwie wartości przesyłane przez master, które zostają zapisane do tablicy. Pierwszą wartością jest komenda, a drugą wartość zapisywana do rejestru adresowanego otrzymaną komendą. Gdy zostaną odebrane dwa bajty danych, zapisanie wartości do dwóch linii wyjściowych portu PB oraz generatora PWM następuje w programie głównym i jest zależne od otrzymanej komendy (pierwszego bajtu). W programie głównym, oprócz zapisu otrzymanych od układu master danych, następuje również odczyt wartości z przetwornika A/C i konwersja odczytanej wartości 10-bitowej do postaci 8-bitowej. Jak można się przekonać, program układu slave nie jest zbyt skomplikowany i można go w prosty sposób rozbudować, dostosowując do własnych potrzeb. W przykładzie z układem slave będzie się on komunikował z prostym układem master, którego schemat pokazano na rys. 3. Do mikrokontrolera został dołączony jedynie wyświetlacz LCD. Do magistrali I<sup>2</sup>C dołączono wymagane rezystory podciągające R1, R2. Potencjometr P1 umożliwia regulację kontrastu wyświetlacza. Na list. 2 przedstawiono przykład programowej realizacji interfejsu I<sup>2</sup>C master. Pierwszą operacją wykonywaną w programie jest odczyt dwóch linii wejściowych układu I<sup>2</sup>C slave. Ich stan jest następnie wyświetlany w pierwszej linii wyświetla-

Tab. 1. Komendy układu slave

Komenda Adres	Odczyt	Zapis
1	PB4, PB5	PB2, PB3
2	ADC	PWM

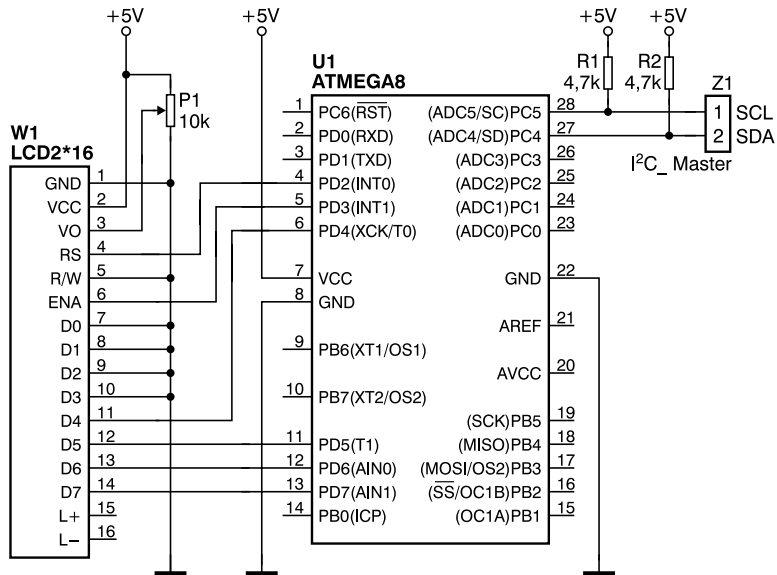
Tab. 2. Adresy układu slave

A0	A1	Adres
0	0	64
1	0	66
0	1	68
1	1	70

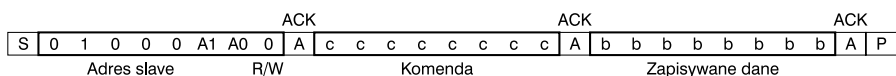
cza LCD. Po przeliczeniu, stan linii wejściowych jest wysyłany z powrotem do układu slave i tu jest zapisywany do dwóch linii wyjściowych. Naciśnięcie któregoś z przycisków S1 lub S2 w układzie slave będzie więc powodować zaświecenie diod LED dołączonych do wyjść układu I<sup>2</sup>C slave. W dalszej kolejności w programie odczytywana jest wartość zmierzona przez przetwornik A/C. Po jej wyświetleniu w drugiej linii wyświetlacza, jest ona z powrotem wysyłana do układu slave, gdzie jest zapisywana do rejestru wypełnienia przebiegu PWM. Ustawiając napięcie potencjometrem P1 w układzie slave, można więc regulować jasność diody D3 sterowanej przebiegiem PWM. Opisujący algorytm działania układu master działa w pętli wykonywanej co około 200 ms. Master komunikuje się z układem slave, wykorzystując adres 64. Adres 65 jest używany podczas odczytu wartości z układu slave. Warto zauważyć, że wartość zmiennej *adres\_c* jest adresem komendy zgodnym z tab. 1. Na rys. 4 pokazano format zapisu danych do układu I<sup>2</sup>C slave. Wysyłane są kolejno: znacznik startu, adres układu, kod komendy i wartość komendy. Na rys. 5 pokazano format odczytu danych z układu slave. W tym przypadku najpierw wysyła się znacznik startu, adres slave do zapisu i komendę. Następnie po ponownym wysłaniu znacznika startu wysyłany jest adres odczytu z układu slave i dalej można już odebrać wartość poprzednio zaadresowaną wartością komendy. Odczyt danych z układu slave jest kończony znacznikiem NACK, po którym następuje znacznik stopu. Najmniej znaczący bit adresu układu slave wskazuje, czy dane będą zapisywane, czy odczytywane.

**Sprzętowa realizacja interfejsu I<sup>2</sup>C slave**

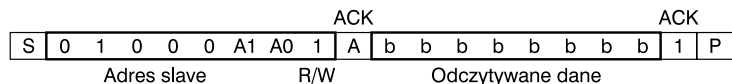
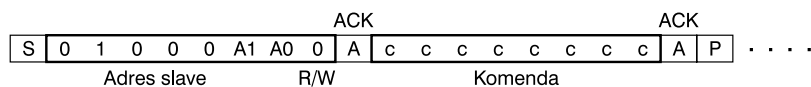
Niektóre większe mikrokontrolery AVR wyposażono w moduł TWI, który jest zgodny ze specyfikacją I<sup>2</sup>C. Układ TWI umożliwia pracę w trybie nadrzędnym oraz podrzędnym, z prędkością transmisji do 400 kbps. W przypadku sprzętowej realizacji interfejsu I<sup>2</sup>C zostanie przedstawiony identyczny przykład działania, jak w poprzednim przykładzie interfejsu I<sup>2</sup>C zrealizowanego programowo. Przy sprzętowej realizacji interfejsu I<sup>2</sup>C slave linie interfejsu I<sup>2</sup>C (w przypadku mikrokontrolera ATmega8) są dołączone do portów PC4 (SD) i PC5 (SC), co zostało zaznaczone na schemacie z rys. 2. Na rys. 6 przedstawiono schemat blokowy sprzętowego interfejsu I<sup>2</sup>C (TWI). W jego skład wchodzi: jednostka kontrolna interfejsu, jednostka adresowa oraz generator prędkości transmisji. Na list. 3 przedstawiono przykład programu realizującego układ I<sup>2</sup>C slave z wykorzystaniem TWI. W Bascomie do tego celu wymagana jest biblioteka *i2c\_twislave.lbx*.



Rys. 3. Przykładowy układ master



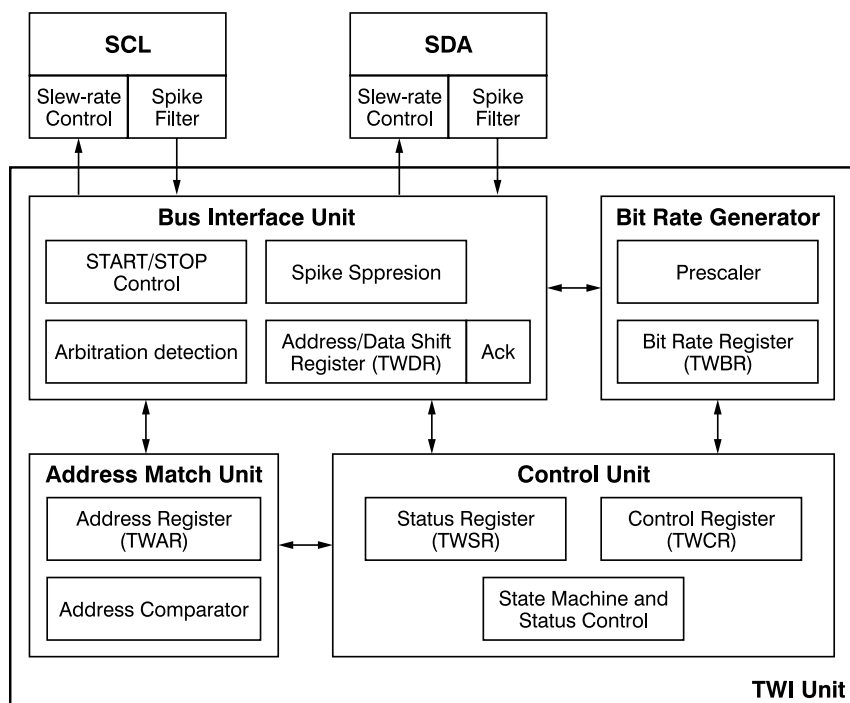
Rys. 4. Format zapisu danych do układu I<sup>2</sup>C slave



Rys. 5. Format odczytu danych z układu slave

Konfiguracja sprzętowego interfejsu TWI tak, aby pracował jako slave, umożliwia komenda *CONFIG TWISLAVE*, której składnia jest następująca:  
**CONFIG TWISLAVE = address , BTR = value , BITRATE = value SAVE=option**

gdzie:  
 Address – parzysty adres układu slave,  
 BTR – liczba odbieranych danych (układ slave będzie oczekiwał na podaną liczbę odbieranych danych),



Rys. 6. Schemat blokowy sprzętowego interfejsu I<sup>2</sup>C (TWI)

BITRATE – wartość częstotliwości zegara (100000 oznacza 100 kHz),

SAVE – informuje kompilator, czy podczas przerwania mają być zapisywane rejestry mikrokontrolera.

Kompilator automatycznie tworzy zmienne *Tw\_i*, *Tw\_i\_btr* i *Tw\_i\_bt看*. Zmienna *Tw\_i* przechowuje otrzymany lub wysłany bajt danych. Zmienna *Tw\_i\_btr* jest indeksem otrzymanych danych. Za jej pomocą można w prosty sposób otrzymywane dane umieszczać w tablicy. Zmienna *Tw\_i\_bt看* może być wykorzystywana do sprawdzania, której wartości potrzebuje master. Ponieważ w TWI wykorzystywane jest przerwanie, należy odblokować globalny system przerw. W przykładowym programie konfiguracja interfejsu TWI slave jest następująca:

```
Config Twislave = 64, Btr = 1, Bitrate = 100000
```

Adres układu slave jest równy 64, oczekiwany będzie jeden bajt danych, a częstotliwość przebiegu zegarowego będzie równa 100 kHz. Adres układu slave jest przechowywany w rejestrze *Twar* i za jego pomocą można zmienić adres układu slave, który w przykładzie jest zależny od linii PC1 i PC2 mikrokontrolera. Gdy master wysyła dane, wywołany jest podprogram *Tw\_i\_gotdata*, w którym odbierane są dwa bajty

danych. Po zdekodowaniu są one następnie w pętli głównej programu zapisywane albo do rejestru PWM, albo linii wyjściowych portu PB – w zależności od komendy. Gdy master odczytuje dane z układu slave, wywołany jest podprogram oznaczony etykietą *Tw\_i\_master\_needs\_byte*. W zależności od zapisanej komendy następuje w nim wysłanie do mastera stanów linii wejściowych portu PB bądź zmierzonej wartości przez przetwornik A/C. Zawartość podprogramów wywołanych przez interfejs TWI jest prawie identyczna jak w przypadku procedur wywołanych przez programowy interfejs I<sup>2</sup>C slave. Sprzętowy interfejs I<sup>2</sup>C slave wywołuje kilka innych podprogramów, które zazwyczaj nie będą wykorzystywane, a o których można się więcej dowiedzieć z pomocy Bascom AVR. Na list. 4 został przedstawiony przykład sprzętowej realizacji układu I<sup>2</sup>C master. W przypadku sprzętowej realizacji interfejsu I<sup>2</sup>C pracującego jako master wymagana jest konfiguracja częstotliwości sygnału zegarowego sprzętowego interfejsu I<sup>2</sup>C (TWI). Nie można również zapomnieć o konfiguracji linii interfejsu I<sup>2</sup>C. Informacją dla kompilatora, że będzie używany sprzętowy interfejs I<sup>2</sup>C, jest załączenie w programie biblioteki *i2c\_twi.lbx*. Do konfiguracji częstotliwości zegara sprzętowego interfejsu

TWI służy instrukcja CONFIG TWI, której parametrem jest częstotliwość linii zegarowej. Wartość 100.000 będzie informować o częstotliwości 100 kHz. Częstotliwość sygnału zegarowego magistrali I<sup>2</sup>C nie powinna być większa niż 400 kHz. Instrukcje wykorzystywane do komunikacji z interfejsem I<sup>2</sup>C slave są identyczne jak w przypadku programowej realizacji interfejsu I<sup>2</sup>C master.

## Podsumowanie

Wykorzystanie interfejsu I<sup>2</sup>C do połączenia ze sobą kilku bloków urządzenia jest wyborem prostym i tanim. W małych mikrokontrolerach AVR z wejściami INT0 i T0, ale niemających interfejsu TWI, można zastosować programowy interfejs I<sup>2</sup>C slave lub master. W większych mikrokontrolerach AVR, choćby z rodziny ATmega, bez problemu można wykorzystać zawarty w nich blok TWI pracujący w konfiguracji slave. Dzięki implementacji interfejsu I<sup>2</sup>C slave w mikrokontrolerach, można zbudować układ podrzędny, który w elastyczny sposób będzie mógł być dostosowany do własnych potrzeb.

Marcin Wiązania, EP  
marcin.wiazania@ep.com.pl

R E K L

**www.elplast.pl**

PROFESJONALNY PRODUCENT:

- miniaturowych zasilaczy impulsowych
- przetwornic DC/AC i DC/DC
- systemów zasilających

"ELPLAST.. Spółka z o.o., ul. Armii Krajowej 9, 58-100 Świdnica  
tel. 074/ 853-34-72, tel./fax 074/ 852-38-20, e-mail: info@elplast.pl

PŁYTKI DRUKOWANE

**SATLAND**  
PROTOTYPE

Szukasz profesjonalnego producenta PCB?  
Masz nietypowy projekt, a może zależy Ci na czasie?  
Właśnie znalazłeś najlepsze rozwiązanie!

**JESTEŚMY JEDYNĄ W POLSCE FIRMĄ REALIZUJĄCĄ  
ZAMÓWIENIA W 5 GODZIN!**

**EKSPRESOWO  
PROFESJONALNIE  
TERMINOWO  
KONKURENCYJNE CENY**

Ceny już od 10 zł/dm<sup>2</sup>

**www.prototypy.com**  
Siedziba firmy: ul. Sarnia 5, 80-336 Gdańsk tel. (058) 554-07-64

A M A

**Qwerty**  
www.qwerty.pl

**KLAWIATURY,  
ELEWACJE,  
TABLICZKI  
I ZESTYKI FOLIOWE**

- ▶ PROJEKTUJEMY
- ▶ PRODUKUJEMY
- ▶ SPRZEDAJEMY

Towarzystwo Elektrotechnologiczne Qwerty Sp. z o.o.  
ul. Siewna 21, 94-250 Łódź,  
e-mail: qwerty@qwerty.pl; www.qwerty.pl;  
tel. 042 632 47 92, 633 32 84, 630 42 64, fax 042 632 85 93