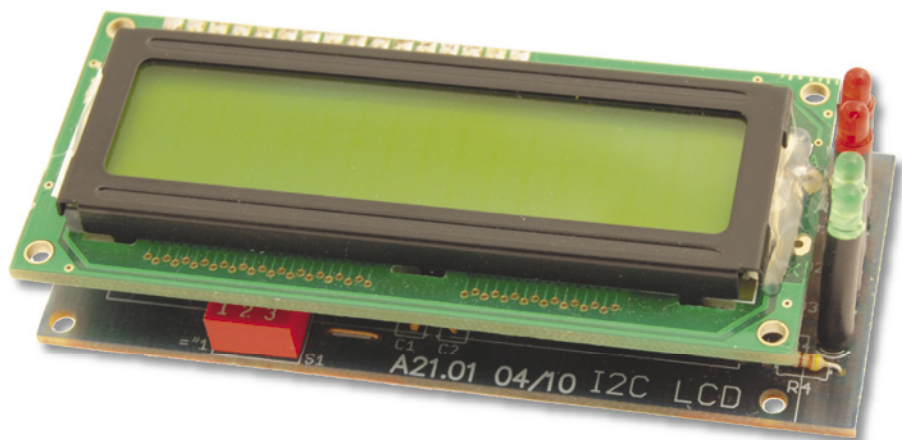




Wyświetlacz LCD sterowany magistralą I²C

W interfejsach użytkownika urządzeń elektronicznych coraz częściej są stosowane wyświetlacze graficzne. Mimo to, popularność alfanumerycznych wyświetlaczy LCD nie maleje, zwłaszcza w prostszych aplikacjach. Typową magistralę, przez którą wyświetlacz komunikuje się z systemem mikroprocesorowym stanowią 3 linie sterujące i 4 lub 8 linii danych. Czasami są to wymagania zbyt duże.

Rekomendacje: moduł typowego wyświetlacza alfanumerycznego, który można zastosować, gdy w systemie mikroprocesorowym brakuje linii do obsługi interfejsu.



Ciekotkryształiczne wyświetlacze alfanumeryczne są już od dawna standardowym elementem interfejsu użytkownika w wielu systemach mikroprocesorowych. Z pewnością do popularyzacji tego typu elementów przyczyniła się firma Hitachi opracowując wiele lat temu standard sterownika HITACHI HD44780, który stał się esencją tego typu sterowników w wykonaniach różnych producentów. Pomimo swojej prostoty, elastyczność zastosowanych rozwiązań zapewniła mu prawdziwą „nieśmiertelność” techniczną, niczym ta, jaką może się poszczycić legendarny NE555. Niezależnie od wprowadzenia od tego czasu wielu nowych magistral danych, producenci wyświetlaczy LCD nadal wykorzystują rozwiązanie zaproponowane przez firmę Hitachi. Praktyka pokazuje jednak, iż są takie systemy,

w których zachodzi potrzeba „wyniesienia” wyświetlacza poza płytkę sterownika (na niewielką odległość) lub obsłużenia więcej niż jednego elementu tego typu. Wiąże się z tym komplikacja sterowania, jak i samych połączeń. Do takich właśnie zastosowań został opracowany poniższy układ. Umożliwia on sterowanie pracą alfanumerycznego wyświetlacza LCD 2x16 znaków za pośrednictwem magistrali I²C, przy okazji wprowadzając szereg dodatkowych funkcji takich jak:

- programowe ustawianie kontrastu wyświetlacza,
- programowe ustawianie jasności podświetlenia matrycy LCD,
- płynne wygaszanie i załączanie podświetlenia matrycy LCD,
- sprzętowe ustawianie adresu wyświetlacza w przestrzeni adresowej magistrali I²C,
- możliwość podłączenia do ośmiu wyświetlaczy na wspólnej magistrali I²C,
- programowe sterowanie dodatkowymi czterema diodami LED,
- wyświetlanie zmiennych liczbowych typu bajt,
- wyświetlanie zmiennych liczbowych typu word,
- funkcja testu matrycy LCD,
- predefiniowane polskie znaki diakrytyczne (małe litery),
- pamięć parametrów kontrastu i jasności podświetlenia matrycy LCD.

Dodatkowym atutem prezentowanego układu jest fakt, iż płytkę drukowaną ma wymiar i rozmieszczenie otworów montażowych typowego wyświetlacza LCD 2x16 znaków. Została zaprojektowana w taki sposób, aby mogła być przykręcona do wyświetlacza stanowiąc jeden

AVT-5154

W ofercie AVT:
AVT-5154A – płytkę drukowaną
AVT-5154B – płytkę + elementy

PODSTAWOWE PARAMETRY

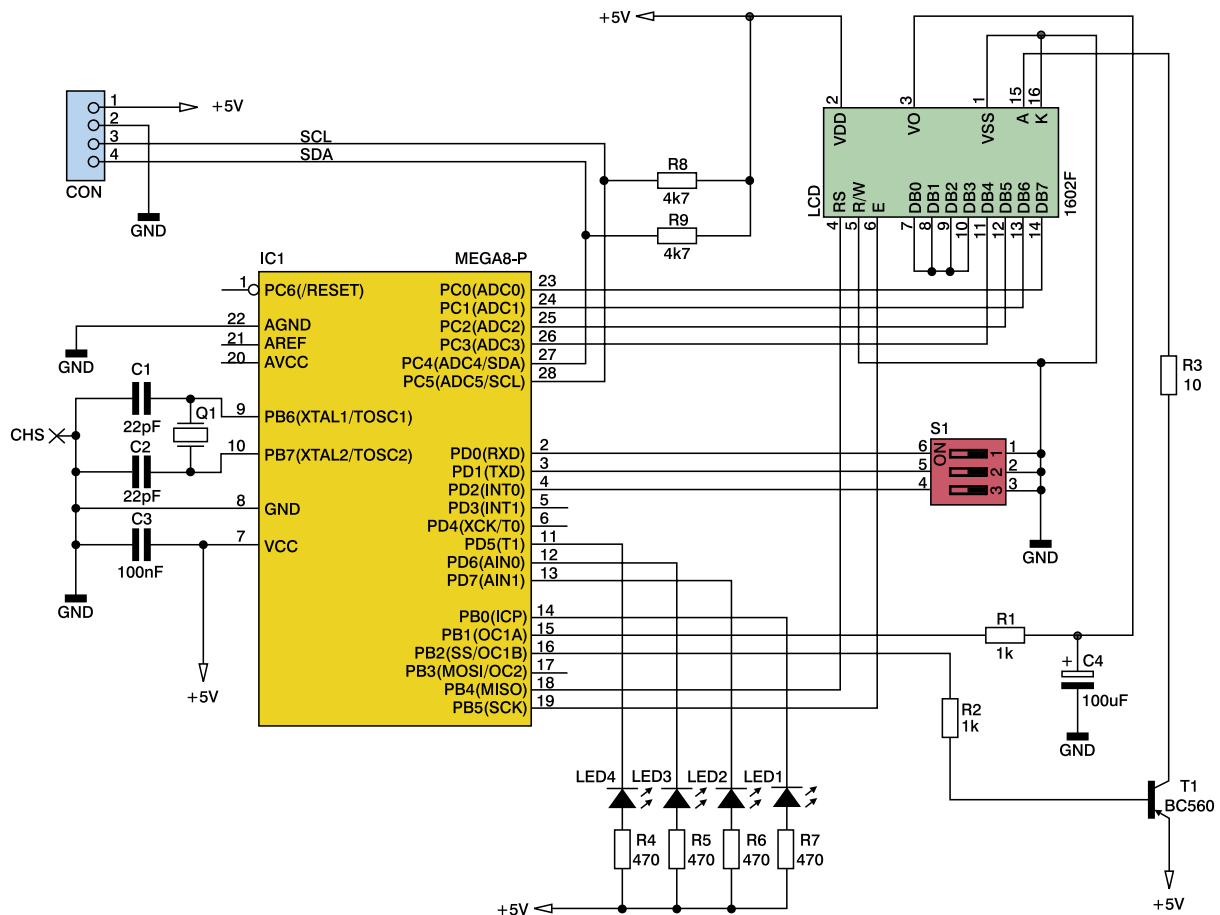
- | | |
|---|---|
| • Płytkę o wymiarach 85x36 mm | 400 kHz |
| • Napięcie zasilania: 5 V | • Liczba dostępnych rozkazów: 18 |
| • Maksymalny prąd obciążenia: 140 mA | • Gotowość po włączeniu zasilania: 350 ms |
| • Maksymalna częstotliwość magistrali I ² C: | • Liczba dostępnych adresów I ² C: 8 |



PROJEKTY POKREWNE

wymienione artykuły są w całości dostępne na CD

Tytuł artykułu	Nr EP/EdW	Kit
Wyświetlacz siedmio-segmentowy JUMBO	EdW 4/1997	AVT-2222
Wielkogabarytowy wyświetlacz siedmio-segmentowy	EP 7/1998	AVT-1186
Podwójny wyświetlacz siedmio-segmentowy sterowany I ² C	EP 8/2000	AVT-859
8-cyfrowy wyświetlacz LED z interfejsem SPI	EP 6/2006	AVT-934
Moduł wyświetlacza LED sterowany magistralą I ² C	EP 9/2008	AVT-5147
Moduł wyświetlacza LED z interfejsem I ² C	EP 3/2000	AVT-1263



Rys. 1. Schemat ideowy interfejsu

zwarty i niewielki moduł. Urządzenie to znajdzie również zastosowanie w systemach, w których nie można byłoby zastosować zwykłego wyświetlacza LCD z uwagi na brak wolnych portów mikrokontrolera.

Dla przyjętych założeń konstrukcyjnych optymalnym rozwiązaniem układowym okazało się zastosowanie mikrokontrolera ATmega8. Posiada on w swojej strukturze sprzętowy generator PWM oraz sprzętowy interfejs TWI, który jest dokładnym odpowiednikiem interfejsu I²C firmy Philips. Wydawać by się mogło, iż zastosowanie tego mikrokontrolera jest nadmierowe z uwagi na możliwości jakie oferuje, jednak przy obecnych, bardzo niskich cenach mikrokontrolerów, jest to wybór ze wszystkich stron optymalny. Schemat układu pokazano na rys. 1.

Jak widać nasz układ jest bardzo prostym sterownikiem zbudowanym z minimalnej liczby niezbędnych elementów. Obwód oscylatora Q1, C1, C2 jest rozwiązaniem opcjonalnym, gdyż może być zastąpiony przez wewnętrzny oscylator mikrokontrolera ATmega8 – wówczas należy odpowiednio ustawić tzw. *Fuse bity* mikrokontrolera. Przelicznik S1 służy do sprzętowego ustawienia adresu urządzenia na magistrali I²C, przy czym adres ten jest wyznaczany w sposób przedstawiony niżej.

A3, A2, A1 to segmenty przelicznika typu

MSB						LSB
1	0	0	0	A3	A2	A1
						0

dip-switch podłączone odpowiednio do portów mikrokontrolera PD2, PD1, PD0. Pozycja załączona (ON) oznacza wyzerowanie odpowiedniego bitu, zaś pozycja wyłączona (OFF) powoduje ustawienie wybranego bitu. Port PB1, będący jednocześnie wyjściem OC1A zintegrowanego, sprzętowego generatora PWM (a dokładnie licznika–czasomierza Timer1 pracującego w trybie generatora PWM) wykorzystany jest do sterowania kontrastem matrycy wyświetlacza LCD. Jest to możliwe dzięki zastosowaniu prostego układu całkującego R1–C4, który z uwagi na bardzo mały prąd obciążenia dla źródła napięcia kontrastu, nie musi być wyposażony we wtórnik napięciowy. Dla tego typu układu, napięcie na wyprowadzeniu Vo wyświetlacza LCD jest wprost proporcjonalne do wypełnienia impulsów sygnału PWM generowanego na wyprowadzeniu PB1 mikrokontrolera (regulacja w 255 krokach). Drugi kanał (OC1B) wbudowanego generatora PWM (wyprowadzenie PB2) jest wykorzystywany do regulacji jasności podświetlenia matrycy LCD, korzystając z zależności, iż jasność świecenia diod LED podświetlacza jest wprost proporcjonalna do wypełnienia impulsów sygnału 2 kanału generatora PWM. Rezystor R3 jest elementem opcjonalnym, ograniczającym prąd diod LED podświetlacza – zwykle rezystor ten jest fabrycznie montowany na płycie wyświetlacza (jako element SMD). Dodatkowo przewidziano możliwość sterowania czterema opcjonalnymi diodami LED (LED1...LED4).

W tym celu wykorzystano kolejne cztery porty mikrokontrolera ATmega8.

W tym miejscu należałoby krótko opisać sprzętowy interfejs I²C, który jak już wiemy przez firmę Atmel został nazwany TWI (dwuprzewodowy interfejs szeregowy). Jego wybrane cechy to:

- 4 główne tryby pracy (Master jako nadajnik, Master jako odbiornik, Slave jako nadajnik i Slave jako odbiornik),
- ustawiany, sprzętowy 7-bitowy adres w przestrzeni adresowej magistrali I²C,
- tryb multi-master z obsługą arbitrażu,

WYKAZ ELEMENTÓW

Rezystory

R1, R2: 1 kΩ
R3: 10 Ω
R4...R7: 470 Ω
R8, R9: 4,7 kΩ

Kondensatory

C1, C2: ceramiczny 22 pF
C3: ceramiczny 100 nF
C4: 100 μF/16 V

Półprzewodniki

IC1: ATmega8

T1: BC560B

LED1...LED4: dioda LED 3 mm

DISPLAY LCD: wyświetlacz LCD 2x16 typ HY-1602

Inne

Q1 – rezonator kwarcowy 8 MHz (niski)

S1 – przelicznik dip-switch 3-pozycyjny

CON – złącze goldpin 4-pin



- szybkość transmisji do 400 kbit/s,
- wbudowany reduktor zakłóceń magistrali,
- obsługa wywołania ogólnego (*General Call*),
- obsługa interfejsu z trybów uśpienia mikrokontrolera.

Komunikacja z interfejsem TWI mikrokontrolera odbywa się poprzez 5 rejestrów sterujących, których krótki opis i znaczenie poszczególnych bitów przedstawiono poniżej.

Rejestr TWBR (rys. 2) ustala stopień podziału częstotliwości zegara mikrokontrolera dla uzyskania sygnału zegarowego (SCL) magistrali I²C – wykorzystywany jedynie dla trybu pracy jako Master. Częstotliwość tę obliczamy według wzoru:

$$SCL = \frac{CPU}{16 + 2(TWBR) \cdot 4^{TWPS}}$$

gdzie:

TWPS – bity preskalera (rejestr TWSR).

Rejestr TWCR (rys. 3) umożliwia sterowanie pracą interfejsu TWI. Poszczególne bity rejestru TWCR mają następujące znaczenie (i odpowiadające im funkcje):

TWINT (Bit7) – Flaga przerwania ustawiana przez sprzęt (interfejs TWI) po wystąpieniu następujących zdarzeń:

- w trybie Master po transmisji przez układ TWI sygnału Start lub „powtórzony” Start,
- w trybie Master po transmisji przez układ TWI adresu układu Slave,
- w trybie Master po transmisji przez układ TWI bajtu danych,
- w trybie Master po nieudanym arbitrażu,
- w trybie Slave po zaadresowaniu własnym adresem lub wywołaniem ogólnym,
- w trybie Slave po odbiorze bajtu danych,
- w trybie Slave po odbiorze sygnału Stop lub „powtórzony” Start,
- przy błędzie magistrali.

W przypadku, gdy włączono globalny system przerwania oraz zezwolono na przerwanie od układu TWI (bit TWIE w rejestrze TWCR jest ustawiony) mikrokontroler wykona skok do procedury obsługi przerwania od układu TWI. Należy pamiętać, iż flaga ta nie jest kasowana automatycznie i wymaga skasowania (wpisanie logicznej jedynki) przez program użytkownika – do tego czasu układ TWI będzie „ściągał” sygnał zegarowy (SCL) do poziomu logicznego zera, nie pozwalając tym samym na przyjęcie nadchodzących informacji (wstrzymując transmisję I²C). Należy także zauważyć, iż dostęp do rejestrów TWAR, TWSR i TWDR musi być zakończony przed wyzerowaniem tejże flagi, w przeciwnym wypadku układ TWI wygeneruje błąd (ustawi bit TWWC).

TWEA (Bit6) – Bit ten kontroluje proces generowania sygnału potwierdzenia ACK, który może zostać wygenerowany w następujących przypadkach:

- w trybie Slave po odbiorze własnego adresu,
- w trybie Slave po odbiorze sygnału wywołania ogólnego (jeśli bit TWGCE w rejestrze TWAR jest ustawiony),
- w trybie odbiornika (Master lub Slave) po odbiorze bajtu danych.

Wyzerowanie bitu TWEA powoduje, iż układ TWI nie będzie potwierdzał odbieranych danych.

TWSTA (Bit5) – Ustawienie tego bitu powoduje (po zwolnieniu magistrali) wygenerowanie przez układ TWI sygnału Start. Bit ten musi być wyzerowany przez program użytkownika po wygenerowaniu żądanego sygnału Start.

TWSTO (Bit4) – Ustawienie tego bitu w trybie Master powoduje automatyczne wygenerowanie sygnału Stop i jego sprzętowe wyzerowanie. W trybie Slave bit ten może być wykorzystany w celu sygnalizacji układowi Master błędu po

stronie odbiornika i przerwania nadawania (układ Slave zwalnia wtedy magistralę przechodząc w stan wysokiej impedancji).

TWWC (Bit3) – Bit kolizji zapisu – ustawiany w przypadku próby dostępu do rejestru TWDR w czasie, gdy bit TWINT (flaga przerwania) jest wyzerowany.

TWEN (Bit2) – Bit uaktywnienia układu TWI. Gdy jest ustawiony (logiczna jedynka), układ TWI przejmuje kontrolę nad portami mikrokontrolera przeznaczonymi dla obsługi interfejsu TWI (PC4 i PC5).

Bit1 – Zarezerwowany i niewykorzystywany przez użytkownika.

TWIE (Bit0) – Uaktywnia przerwanie od układu TWI. Rejestr TWSR (rys. 4) zawiera stan układu TWI.

TWS7...TWS3 (Bit7...Bit3) – 5 najstarszych bitów rejestru TWSR zawiera status określający bieżący stan układu TWI. Informacja ta jest wykorzystywana przez procedurę obsługi przerwania od TWI w celu podjęcia określonych działań po stronie programu użytkownika. Dla określenia statusu układu TWI należy posłużyć się odpowiednią tablicą kodów statusu i odpowiadających im stanów pracy układu TWI – różnych w zależności od trybu pracy interfejsu. Należy maskować dwa najmłodsze bity rejestru TWSR, aby uniezależnić się od bitów określających ustawienia preskalera częstotliwości sygnału SCL.

Bit2 – Zarezerwowany i niewykorzystywany przez użytkownika.

TWPS1...TWPS0 (Bit1...Bit0) – Bity określające ustawienia preskalera częstotliwości sygnału SCL (patrz: rejestr TWBR). Wartości preskalera określamy na podstawie poniższej tab. 1.

Rejestr TWDR (rys. 5) – rejestr danych.

W trybie nadajnika do rejestru TWDR należy wpisać bajt, który ma być wysłany magistralą I²C.

Bit	7	6	5	4	3	2	1	0
Oznaczenie	TWBR7	TWBR6	TWBR5	TWBR4	TWBR3	TWBR2	TWBR1	TWBR0
Odczyt(R)/Zapis(W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość domyślna	0	0	0	0	0	0	0	0

Rys. 2. Rejestr TWBR (rejestr prędkości transmisji)

Bit	7	6	5	4	3	2	1	0
Oznaczenie	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Odczyt(R)/Zapis(W)	R/W	R/W	R/W	R/W	R	R/W	R	R/W
Wartość domyślna	0	0	0	0	0	0	0	0

Rys. 3. Rejestr TWCR (rejestr sterujący interfejsu TWI)

Bit	7	6	5	4	3	2	1	0
Oznaczenie	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0
Odczyt(R)/Zapis(W)	R	R	R	R	R	R	R/W	R/W
Wartość domyślna	1	1	1	1	1	0	0	0

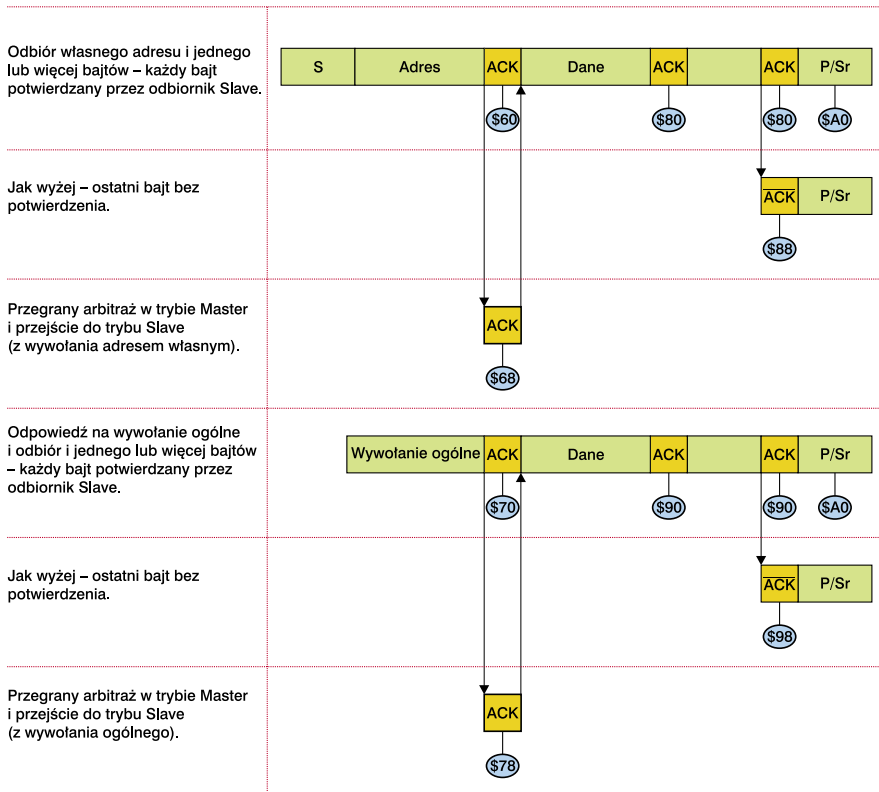
Rys. 4. Rejestr TWSR (rejestr stanu układu TWI)

Bit	7	6	5	4	3	2	1	0
Oznaczenie	TWD7	TWD6	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0
Odczyt(R)/Zapis(W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość domyślna	1	1	1	1	1	1	1	1

Rys. 5. Rejestr TWDR (rejestr danych)

Bit	7	6	5	4	3	2	1	0
Oznaczenie	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
Odczyt(R)/Zapis(W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość domyślna	1	1	1	1	1	1	1	0

Rys. 6. Rejestr TWAR (rejestr adresu układu TWI)

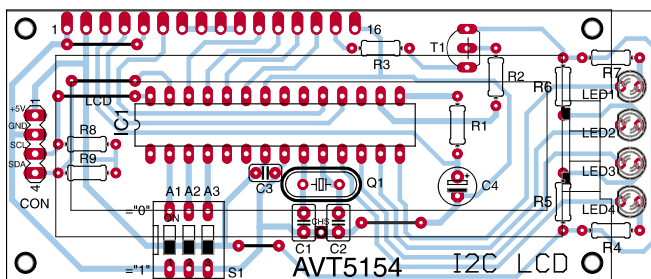


Legenda:

- Akcja inicjowana przez układ Master
- Akcja inicjowana przez układ Slave
- Status stanu pracy układu TWI odczytywany z rejestru stanu TWSR (bity preskalera maskowane)

S – sygnał Start
 Sr – powtórzony sygnał Start
 P – sygnał Stop
 ACK – potwierdzenie odbiornika
 ACK – bez potwierdzenia odbiornika

Rys. 7. Możliwe stany pracy magistrali I²C oraz stany pracy układu TWI dla trybu Slave odbiornika



Rys. 8. Schemat montażowy sterownika

W trybie odbiornika rejestr TWDR zawiera ostatnio odebrany bajt. Dostęp do rejestru TWDR jest możliwy po sprzętowym ustawieniu flagi TWINT przez układ TWI i do czasu jej wyczyszczenia przez program użytkownika (np. w procedurze

Tab. 1. Ustawianie wartości preskalera za pomocą bitów TWPSx

TWPS1	TWPS0	Wartość preskalera
0	0	1
0	1	4
1	0	16
1	1	64

Tab. 2. Możliwe wartości rejestru stanu TWSR dla pracy odbiornika w trybie Slave i odpowiadające im stany układu TWI

TWSR	Stan interfejsu TWI
0x60	Odebrano własny adres i potwierdzono go sygnałem ACK
0x68	Po nieudanym arbitrażu (jako Master) odebrano własny adres i potwierdzono go sygnałem ACK
0x70	Odebrano wywołanie ogólne i potwierdzono je sygnałem ACK
0x78	Po nieudanym arbitrażu (jako Master) odebrano sygnał wywołania ogólnego i potwierdzono go sygnałem ACK
0x80	Poprzednio zaadresowany układ odebrał bajt danych i potwierdził go sygnałem ACK
0x88	Poprzednio zaadresowany układ odebrał bajt danych i NIE potwierdził go sygnałem ACK (NACK)
0x90	Poprzednio zaadresowany układ z wywołania ogólnego odebrał bajt danych i potwierdził go sygnałem ACK
0x98	Poprzednio zaadresowany układ z wywołania ogólnego odebrał bajt danych i NIE potwierdził go sygnałem ACK (NACK)
0xA0	Odebrano sygnał Stop lub „powtórzony” Start

przerwania od układu TWI). Z tej właściwości wynika również fakt, iż pierwszy dostęp do rejestru TWDR jest możliwy dopiero po wystąpieniu pierwszego przerwania od układu TWI, czyli po pierwszym ustawieniu bitu TWINT przez sprzęt. W przypadku wyprowadzenia mikrokontrolera ze stanu uśpienia wywołanego zdarzeniem magistrali I²C (i wywołaniem przerwania od TWI), stan rejestru TWDR jest nieustalony (rejestr może zawierać wartość przypadkową). Generowanie dodatkowego taktu sygnału zegarowego potrzebnego dla potwierdzenia otrzymania bajtu przez odbiornik i kontroli potwierdzenia odebranego bajtu (w trybie nadawnika) lub potwierdzenia odebranego bajtu (w trybie odbiornika) dokonywane jest automatycznie przez układ TWI zależnie od ustawień bitu TWEA.

Rejestr TWAR (rys. 6) zawiera adres układu TWI.

TWA6...TWA0 (Bit7...Bit1) – 7 najstarszych bitów rejestru TWAR zawiera 7-bitowy adres układu TWI w przestrzeni adresowej magistrali I²C w przypadku trybu pracy jako Slave.

TWGCE (Bit0) – Gdy jest ustawiony, zezwala na obsługę przez układ TWI tzw. wywołania ogólnego, czyli zaadresowania magistrali I²C adresem 0x00.

Dzięki temu, że układ TWI został zaprojektowany tak, aby mógł generować przerwania, w których program użytkownika dokonuje odpowiedniej reakcji, mikrokontroler jest zwolniony z konieczności ciągłego sprawdzania stanu magistrali I²C. Zazwyczaj w procedurze obsługi przerwania od układu TWI sprawdzany jest status interfejsu poprzez sprawdzenie rejestru TWSR i podejmowana jest akcja programu użytkownika polegająca na odpowiednich modyfikacjach rejestru TWCR i/lub TWDR. Sposób obsługi układu TWI, a co za tym idzie paleta dostępnych akcji po stronie programu użytkownika zależy w dużej mierze od trybu, w jakim on pracuje. Dla naszego układu wyświetlacza LCD sterowanego magistralą I²C układ TWI mikrokontrolera ATmega8 pracuje jako Slave wyłącznie w trybie odbiornika. Możliwe stany pracy magistrali I²C oraz stany pracy układu TWI dla tego trybu przedstawiono na rys. 7.

Tab. 3. Opis funkcji w przykładowej aplikacji

Numer funkcji	Opis	Dane	Znaczenie danych
0x00	Wyświetla znaki	Dana(1)...Dana(40)	Wyświetla znaki, których kody ASCII zawierają zmienne Dana(x), począwszy od bieżącej pozycji kursora w bieżącym wierszu (maks. 40 znaków)
0x01	Czyści ekran	Brak (nie dotyczy)	Po wyczyszczeniu ekranu kursor ustawiany jest na początku górnego wiersza
0x02	Ustawia parametry ekranu	Dana(1)	Bit 0 – włączenie („1”)/wyłączenie („0”) migania kursora Bit 1 – włączenie („1”)/wyłączenie („0”) kursora Bit 2 – włączenie („1”)/wyłączenie („0”) ekranu
0x03	Parkuje kursor	Dana(1)	Bit 0 – ustawia kursor na początku dolnego („1”)/górnego („0”) wiersza
0x04	Przestawia kursor	Dana(1)...Dana(2)	Przestawia kursor do wiersza Dana(1) na pozycję Dana(2)
0x05	Przesuwa kursor	Dana(1)	Bit 0 – przesuwają pozycję kursora o jedną pozycję w prawo („1”)/lewo („0”)
0x06	Przesuwa zawartość ekranu	Dana(1)	Bit 0 – przesuwają zawartość ekranu o jedną pozycję w prawo („1”)/lewo („0”)
0xA0	Definiuje znak pamięci CGRAM ⁽¹⁾	Dana(1)...Dana(9)	Dana(1) – numer znaku (0...7) pamięci CGRAM Dana(2)...Dana(9) – definicja znaku. Po wykonaniu tej funkcji kursor jest ustawiany na początku górnego wiersza.
0xA1	Wyświetl liczbę typu Bajt	Dana(1)	Wyświetla zmienną typu Bajt jako liczbę – bez zer nieznaczących
0xA2	Wyświetl liczbę typu Word	Dana(1)...Dana(2)	Wyświetla zmienną typu Word jako liczbę – bez zer nieznaczących, przy czym Dana(1) to starszy bajt zmiennej typu Word, a Dana(2) to bajt młodszy
Funkcje specjalne			
0x64	Ustawia kontrast wyświetlacza ⁽²⁾	Dana(1)	Ustawia kontrast wyświetlacza z jednoczesnym zapisaniem wartości do nieulotnej pamięci EEPROM jako wartości startowej po włączeniu zasilania (zakres 0...255)
0x65	Ustawia jasność podświetlenia wyświetlacza	Dana(1)	Ustawia jasność podświetlenia wyświetlacza z jednoczesnym zapisaniem wartości do nieulotnej pamięci EEPROM jako wartości startowej po włączeniu zasilania (zakres 0...255)
0x66	Włącz diodę LED	Dana(1)	Włącza diodę LED o numerze Dana(1)
0x67	Wyłącz diodę LED	Dana(1)	Wyłącza diodę LED o numerze Dana(1)
0x68	Sterowanie grupowe LED	Dana(1)	Bit 0 – włączenie („0”)/wyłączenie („1”) diody LED 1 Bit 1 – włączenie („0”)/wyłączenie („1”) diody LED 2 Bit 2 – włączenie („0”)/wyłączenie („1”) diody LED 3 Bit 3 – włączenie („0”)/wyłączenie („1”) diody LED 4
0x69	Wyłącz podświetlenie wyświetlacza – stopniowo	Brak (nie dotyczy)	Wyłącza (w sposób płynny, w czasie 1 sek.) podświetlenie wyświetlacza
0x6A	Włącz podświetlenie wyświetlacza – stopniowo	Brak (nie dotyczy)	Włącza (w sposób płynny, w czasie 1 sek.) podświetlenie wyświetlacza – do wartości zapisanej w pamięci EEPROM
0xFF	Test	Brak (nie dotyczy)	Test wyświetlacza – zapalone wszystkie punkty
Uwagi: 1) standardowo, po włączeniu zasilania, do pamięci CGRAM wyświetlacza LCD ładowane są dane umożliwiające wyświetlenie małych, polskich liter. Adresy tych znaków oraz odpowiadające im znaki wyglądają następująco (adres-znak): 0x00-ą, 0x01-ć, 0x02-ę, 0x03-ł, 0x04-ń, 0x05-ó, 0x06-ś, 0x07-ż. 2) typowy, optymalny kontrast wyświetlacza wynosi około 215 i jest zależny od modelu wyświetlacza.			

Możliwe wartości rejestru stanu TWSR dla powyższego przypadku i odpowiadające im stany układu TWI przedstawiono w tab. 2 (warto-

ści podano przy założeniu maskowania bitów preskalera (TWPS1 i TWPS0).

Jako, że specyfikacja magistrali I²C określa

jedynie elektryczne i logiczne jej właściwości, w każdym przypadku implementacji dla konkretnej aplikacji układowej zachodzi potrzeba zaprojektowania ramki transmisji w zależności od technicznych założeń projektu. Dla naszego projektu wygląda ona następująco:

Start – Adres^{ACK} – Numer Funkcji^{ACK}
 – Dana(1)^{ACK} ... Dana(x)^{ACK} – Stop

przy czym liczba wymaganych danych jest uzależniona od wybranej funkcji urządzenia (tab. 3).

Należy podkreślić, iż każdy z przesyłanych bajtów musi zostać potwierdzony przez układ odbiornika (w naszym przypadku jest to poniższy sterownik), a co za tym idzie Master (jako nadajnik) musi wygenerować dodatkowy cykl zegarowy sygnału SCL przeznaczony na potwierdzenie (ACK) po stronie odbiornika. W przypadku niekompletnej ramki danych lub nieznannej funkcji, na wyświetlaczu LCD zostanie wyświetlony komunikat o błędzie, powodując jednocześnie utratę zawartości ekranu. Przesłanie danych otrzymanych magistralą I²C do sterownika wyświetlacza LCD następuje w chwili wygenerowania sygnału Stop w przesyłanej ramce transmisji. Należy pamiętać, iż sam wyświetlacz LCD potrzebuje czasu na wykonanie każdej z operacji – czasy wykonania poszczególnych rozkazów zawiera nota aplikacyjna sterownika firmy Hitachi. Sama transmisja danych magistralą I²C nie wpływa w sposób znaczący na wydłużenie tych czasów z wyjątkiem korzystania z funkcji 0x00 (wyświetla znaki) i przesyłania maksymalnej długości tekstu do pamięci DDRAM sterownika ekranu (40 znaków). Tego typu transmisja, przy założeniu częstotliwości sygnału SCL równej 100 kHz, zajmie ok. 3,7 ms. Do czasu zakończenia realizacji każdej z funkcji magistrala I²C pozostaje nieaktywna.

Montaż

Schemat montażowy płytki sterownika przedstawiono na rys. 8. Montaż układu należy rozpocząć od wlutowania zworek, następnie: rezystory, kondensatory (należy zwrócić uwagę na typ elementów: ceramiczny lub elektrolityczny), złącza i podstawki, a na końcu półprzewodniki. Metalową obudowę rezonatora kwarcowego należy połączyć z masą układu – przewidziano odpowiednie wyprowadzenie. Układ należy zasilić napięciem stałym 5 V. Diody LED należy umieścić na odpowiedniej wysokości stosując plastikowe elementy dystansowe. Wyświetlacz LCD montujemy wykorzystując złącze goldpin typu gniazdo-wtyk. Podczas pierwszego uruchomienia należy, w odróżnieniu od zwykłego wyświetlacza LCD, ustawić kontrast wyświetlacza LCD używając do tego celu magistrali I²C oraz funkcji o numerze 0x64. Ustawienie to zostanie zapamiętane w nieulotnej pamięci sterownika jako domyślne po włączeniu zasilania układu.

Robert Wołgajew, EP
robert.wolgajew@ep.com.pl