

RTC M41T56C64: dokładny chronometr z pamięcią EEPROM

Układ M41T56C64, produkowany przez firmę STMicroelectronics (www.st.com) jest niebanalnym połączeniem układu RTC M41T56, pamięci EEPROM M24C64 oraz rezonatora kwarcowego 32768 Hz w 18-wyprowadzeniowej obudowie do montażu powierzchniowego. Rozmieszczenie wyprowadzeń układu przedstawiono na **rys. 1**. Aż 9 wyprowadzeń układu (NC, NF i V_{SS}) przeznaczonych jest do połączenia z masą układu, dwa służą do dołączenia napięcia zasilającego (V_{CC}) i baterii (V_{BAT}). Pozostaje więc jedynie 7 wyprowadzeń funkcyjnych. Wyprowadzenia E0...E2 służą do określenia trzech młodszych bitów adresu układu pamięci EEPROM na magistrali I²C – ich przeznaczenie jest identyczne jak w „samodzielnym” układzie M24C64. Również wyprowadzenie /WC jest powiązane z pamięcią EEPROM i również jego funkcja jest identyczna jak w układzie M24C64. Wyprowadzenie FT/OUT pełni podwójną funkcję. Pracując w trybie FT (*frequency test*) stanowi wyjście sygnału zegarowego podzielonego przez 64, czuli na wyjściu tym generowany jest sygnał o częstotliwości 512 Hz, który

Pomiar czasu jest jednym z podstawowych pomiarów wykonywanych we współczesnych urządzeniach elektronicznych. Zegary mamy w telefonach komórkowych, aparatach fotograficznych, mikrofalówkach, odtwarzaczach MP3 itd., itd. W większości powyższych przypadków super dokładność pomiaru czasu nie jest bardzo istotna, ale mimo tego chcielibyśmy, aby punktualność naszego zegara była jak największa, choćby ze względu na unikanie konieczności dokonywania korekty wskazań. Dobrym rozwiązaniem jest więc stosowanie specjalizowanych układów.



można wykorzystać do kalibracji częstotliwości wbudowanego oscylatora, natomiast w trybie OUT wyprowadzenie to pełni funkcję uniwersalnego wyjścia (z otwartym drenem), którego stan określa jeden z bitów rejestrów układu zegara. Schemat blokowy układu M41T56C64 przedstawiono na **rys. 2**. Z punktu widzenia programisty piszącego program współpracujący z układem M41T56C64 składa się on z dwóch niezależnych układów: M41T56 oraz M24C64 pracujących na wspólnej magistrali I²C. Obydwa bloki funkcjonalne zostaną omówione w dalszej części artykułu. Procedury i przykładowe programy ilustrujące artykuł zostały napisane w asemblerze dla mi-

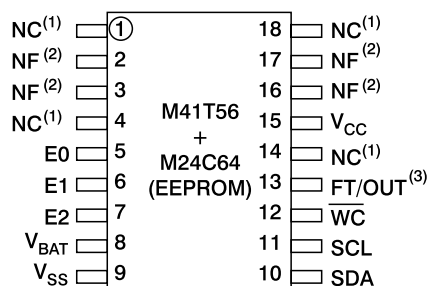
krokontrolera ST7FLITE19 i mogą być uruchomione na komputerku LITEcomp. Schemat podłączenia układu M41T56C64 do LITEcompa przedstawiony jest na **rys. 3**.

Procedury obsługi magistrali I²C

Procedury obsługi magistrali I²C zostały omówione w ramach kursu „LITEcomp – aplikacje” w numerze 6/2007 Elektroniki Praktycznej. Dla przypomnienia publikujemy je na **list. 1**.

Zegar M41T56

Część zegarowa układu M41T56C64 jest funkcjonalnym odpowiednikiem samodzielnego układu M41T56. Posiada on 8 re-

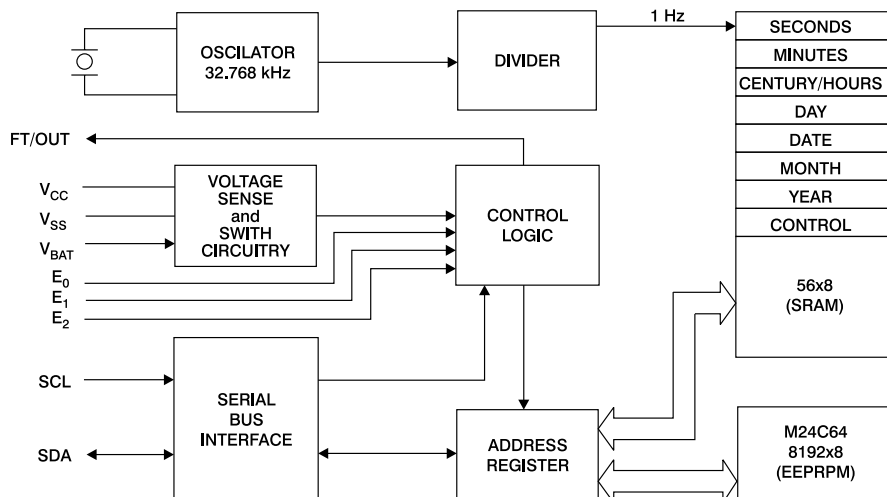


(1) NC – Wyprowadzenia powinny być połączone z masą układu.

(2) NF – Wyprowadzenia powinny być połączone z masą układu, wyprowadzenia 2 i 3 oraz 16 i 17 są wewnętrznie połączone ze sobą

(3) Wyjście z otwartym drenem.

Rys. 1. Rozmieszczenie wyprowadzeń układu M41T56C64



Rys. 2. Schemat blokowy układu M41T56C64

jestrów sterujących służących do ustawiania oraz odczytu informacji o czasie i dacie oraz 56 bajtów pamięci RAM ogólnego przeznaczenia. Strukturę rejestrów układu zegara przedstawiono na **rys. 4**. Informacja o czasie i dacie jest zapisana w formacie BCD. Strukturę bajtu adresu układu M41T56 przedstawiono na **rys. 5**. Adres ten jest ustawiony na stałe i nie ma możliwości jego zmiany, tak więc na jednej magistrali I²C może pracować tylko jeden układ M41T56/M41T56C64.

Zapis danych do rejestrów zegara

Przebiegi na magistrali I²C podczas zapisu danych do rejestrów wewnętrznych zegara oraz pamięci RAM przedstawiono na **rys. 6**. Możliwy jest sekwencyjny zapis kolejnych danych, gdyż adres rejestru wewnętrznego jest automatycznie inkrementowany po każdej operacji zapisu. Kod procedury zapisu danej do rejestru układu M41T56 przedstawiono na **list. 2**.

Odczyt danych z rejestrów zegara

Przebiegi na magistrali I²C podczas odczytu danych z rejestrów wewnętrznych zegara oraz pamięci RAM przedstawiono na **rys. 7**. Podobnie jak w przypadku zapisu, adres rejestru wewnętrznego jest automatycznie inkrementowany po każdej operacji odczytu, przez co możliwy jest sekwencyjny odczyt danych. Procedurę odczytu danych z rejestru układu M41T56 przedstawiono na **list. 3**.

Pamięć EEPROM

Pamięć EEPROM zawarta w układzie M41T56C64 jest funkcjonalnym odpowiednikiem układu pamięci M24C64 zawierającego 8192 bajty pamięci EEPROM. Strukturę bajtu adresu układu pamięci EEPROM przedstawiono na **rys. 8**. Stan bitów E2...E0 jest porównywany ze stanem wypraważeń E2...E0, dzięki czemu na magistrali I²C może pracować do ośmiu układów pamięci EEPROM, np. jeden układ M41T56C64 oraz do siedmiu klasycznych układów M24C64 lub innych układów pracujących na magistrali I²C, których starsza połówka bajtu adresu jest równa 1010.

List. 1. Procedury obsługi magistrali I²C

```

;*****
; Procedura generująca sygnał START
;*****
.I2C_Start
  BRES I2C_DIR, #SDA
  BRES I2C_DIR, #SCL
  BSET I2C_DIR, #SDA      ; wyzerowanie linii SDA
  CALL i2c_delay
  BSET I2C_DIR, #SCL      ; wyzerowanie linii SCL
  RET

;*****
; Procedura generująca sygnał STOP
;*****
.I2C_Stop
  BRES I2C_DIR, #SCL      ; ustawienie linii SCL
  CALL i2c_delay
  BRES I2C_DIR, #SDA      ; ustawienie linii SDA
  RET

;*****
; Procedura zapisująca bajt na magistralę I2C
; wartość bitu ACK zwracana jest we wskaźniku C
; Wysyłany bajt należy umieścić w rejestrze A
;*****
.I2C_Write
  LD i2c_data, A          ; zapamiętanie A w zmiennej i2c_data
  LD A, #8
  LD i2c_counter, A      ; ilość powtórzeń pętli
i2cw1
  BSET I2C_DIR, #SCL      ; stan niski na linii SCL
  BSET I2C_DIR, #SDA      ; stan niski na linii SDA
  LD A, i2c_data
  RLC A                  ; przywrócenie danej
  RLC A                  ; obrót w lewo przez C
  LD i2c_data, A
  JRNC i2cw2            ; zapamiętanie danej
  BRES I2C_DIR, #SDA      ; skok, jeśli C wyzerowany
                          ; w przeciwnym razie ustawienie linii SDA
i2cw2
  BRES I2C_DIR, #SCL      ; ustawienie linii SCL
  DEC i2c_counter        ; zmniejszenie licznika pętli
  JRNE i2cw1            ; skok do i2cw1 jeśli większe od zera
  BSET I2C_DIR, #SCL      ; wyzerowanie linii SCL
  BRES I2C_DIR, #SDA      ; wysoki stan na linii SDA
  BRES I2C_DIR, #SCL      ; wysoki stan na linii SCL
  BTJT I2C_DAT, #SDA, i2cw3 ; stan linii SDA do C
i2cw3
  BSET I2C_DIR, #SDA      ; wyzerowanie linii SDA
  BSET I2C_DIR, #SCL      ; wyzerowanie linii SCL
  RET

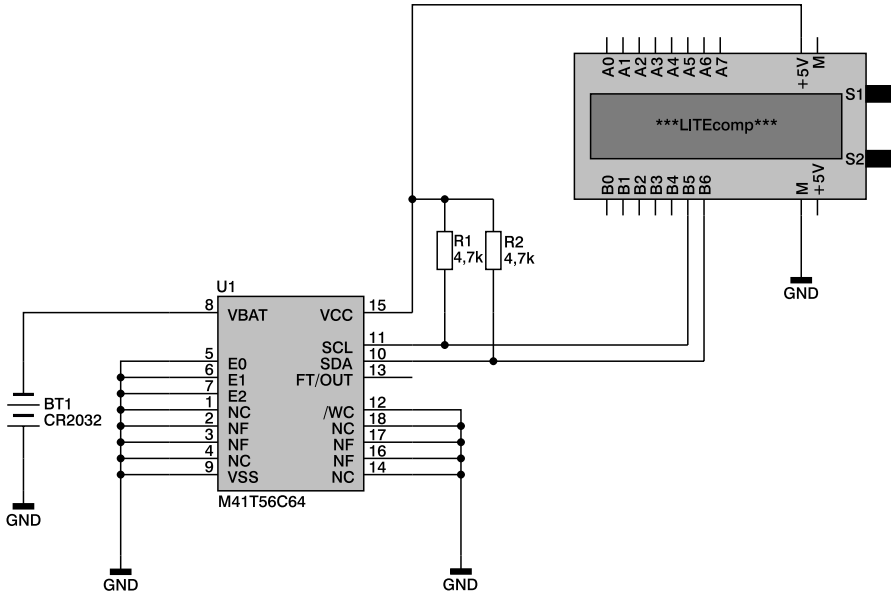
;*****
; Procedura odczytująca bajt z magistrali I2C
; Bit ACK przed wywołaniem procedury należy umieścić we wskaźniku C
;*****
.I2C_Read
  RLC A                  ; zapamiętanie bitu ACK przekazanego poprzez C
  LD i2c_data, A
  LD A, #8
  LD i2c_counter, A      ; ilość powtórzeń pętli
  BRES I2C_DIR, #SDA      ; stan wysoki na linii SDA
i2cr1
  CALL i2c_delay          ; czekamy chwilę
  BRES I2C_DIR, #SCL      ; stan wysoki na linii SCL
  BTJT I2C_DAT, #SDA, i2cr2 ; stan linii SDA do C
i2cr2
  LD A, i2c_data
  RLC A                  ; przywracamy poprzednio odczytane bity
  LD i2c_data, A
  BSET I2C_DIR, #SCL      ; zapamiętujemy nowo odebrany bit
  DEC i2c_counter        ; zapamiętujemy daną
  JRNE i2cr1            ; niski stan na linii SCL
  BSET I2C_DIR, #SDA      ; zmniejszamy licznik pętli
  JRNC i2cr3            ; skok do i2cr1 jeśli większe od zera
  BRES I2C_DIR, #SDA      ; zerujemy linię SDA
  BRES I2C_DIR, #SDA      ; skok do i2cr3 jeśli wskaźnik C wyzerowany
                          ; ustawiamy linię SDA
i2cr3
  BRES I2C DIR, #SCL      ; ustawiamy linię SCL
  BSET I2C_DIR, #SCL      ; zerujemy linię SCL
  BSET I2C_DIR, #SDA      ; zerujemy linię SDA
  LD A, i2c_data
  RET                    ; w A odebrana dana

```

Zapis danych do pamięci EEPROM

Zapis do komórki pamięci jest możliwy na dwa sposoby: zapis pojedynczej komórki pamięci oraz zapis strony o rozmiarze do 32 bajtów. Przebiegi na magistrali podczas zapisu pojedynczego bajtu do pamięci przedstawiono na **rys. 9**. Transmisja jest rozpoczy-

nana od nadania przez układ master sekwencji START oraz adresu układu SLAVE, który jest potwierdzony przez układ pamięci bitem ACK. Następnie transmitowane są dwa bajty składające się na adres komórki pamięci. Każdy z tych bajtów jest potwierdzany przez układ pamięci bitem ACK. Jako ostatni transmitowany jest bajt przeznac-



Rys. 3. Schemat dołączenia układu M41T56C64 do LITECompa

czony do zapisania w pamięci. W zależności od stanu wyprowadzenia /WC bajt ten jest potwierdzany bitem ACK (wyprowadzenie /WC w stanie niskim), bądź też układ pamięci transmituje bit NO-ACK świadczący o niepomyślnym zakończeniu operacji zapisania bajtu do pamięci (wyprowadzenie /WC w stanie wysokim). Procedurę zapisu pojedynczego bajtu danych do pamięci EEPROM przedstawiony jest na list. 4.

Przed wywołaniem procedury należy do zmiennej `m24DstAddr` załadować adres, pod który ma zostać zapisana dana, natomiast w rejestrze A należy umieścić zapisywaną daną.

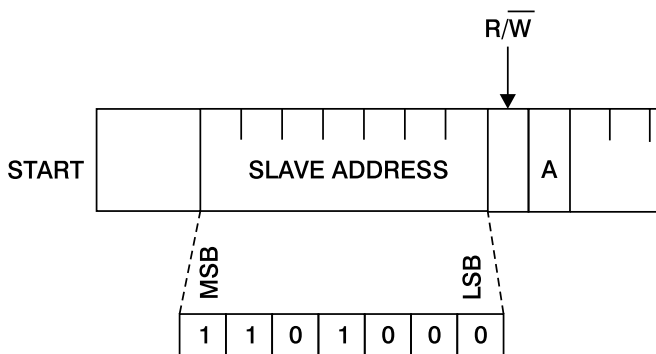
Przebiegi na magistrali podczas zapisu strony 32 bajtów przedstawiono na rys. 10. Transmisja przebiega podobnie, jak w przypadku zapisu jednego bajtu, z tym, że adres, od którego rozpoczynany jest zapis danych powinien być podzielny przez 32. Po nadaniu sekwencji START oraz adresu układu i adresu komórki pamięci układ nadrzędny transmittuje do 32 bajtów przeznaczonych do zapisania w pamięci, które są potwierdzane sygnałem ACK, jeśli wyprowadzenie /WC znajduje się w stanie niskim. Transmisja musi być oczywiście zakończona sygnałem STOP. Procedurę zapisu strony pamięci przedstawiono na list. 5.

Przed wywołaniem procedury, do zmiennej `m24SrcAddr` należy załadować adres początku bufora z danymi przeznaczonymi do zapisania, natomiast do zmiennej `m24DstAddr` adres, pod który dane zostaną zapisane. W rejestrze A należy umieścić rozmiar zapisywanej porcji danych (maksymalnie 32 bajty).

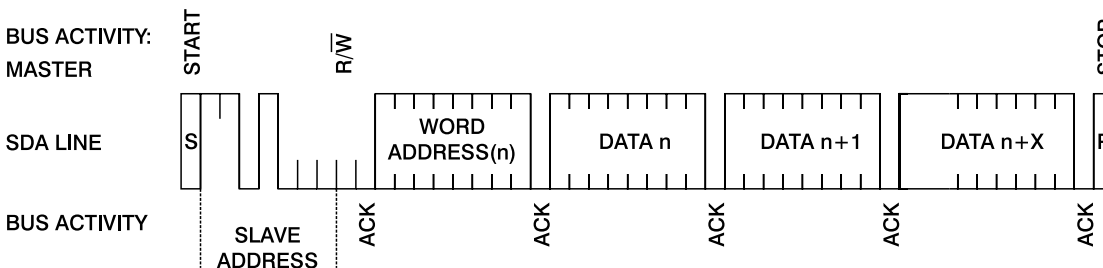
Ponieważ operacja programowania pamięci EEPROM może trwać nawet 5 milisekund, przed zainicjowaniem kolejnej operacji zapisu danych należy odczekać na zakończenie poprzedniej operacji programowania. Jeśli operacja programowania nie została zakończona, przy próbie zaadresowania układu M24C64 odpowie

Address	Data bits								Function/range BCD format	
	D7	D6	D5	D4	D3	D2	D1	D0		
0	ST	10 seconds			Seconds			Seconds	00-59	
1	X	10 minutes			Minutes			Minutes	00-59	
2	CEB ⁽²⁾	CB	10 hours		Hours			Century/hours	01/00-23	
3	X	X	X	X	X	Day		Day	01-07	
4	X	X	10 date		Date			Date	01-31	
5	X	X	X	10 M.	Month			Month	01-12	
6	10 years			Year			Year	00-99		
7	OUT	FT	S	Calibration				Control		

Rys. 4. Struktura rejestrów zegara



Rys. 5. Struktura bajtu adresu zegara



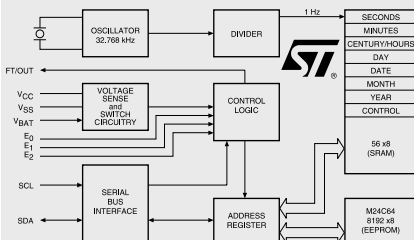
Rys. 6. Przebiegi na magistrali I²C podczas zapisu danych do rejestrów zegara



Niebanalne

RTC

Oferujemy m.in. układ M41T56C64 wyposażony w: RTC z wewnętrznym rezonatorem, pamięć EEPROM (8x8 kb), przełącznik zasilania baterijnego.



W ciągłej sprzedaży także:

- mikrokontrolery 32-b: LPC2000, STM32, STR91x, AVR32, ADuC7, PIC32, AT91,
- mikrokontrolery 8-b: 89C5x, AVR8, EZ-USB, ST7LITE, ST72F, PIC12, PIC16, HC08, HCS08, HCS12
- mostki PHY Ethernet: STE100P, DP83848, ENC28J60,
- gniazda z transformatorami Ethernet,
- układy interfejsowe: LIN, RS232, RS485, CAN, USB, ekspandery I2C, UART, IrDA,
- pamięci: Flash, SRAM, DataFlash, EEPROM, SDRAM (64/128/256 Mb),
- stabilizatory LDO: 1,2; 1,5; 1,8; 3,3 V
- ekspandery I/O z interfejsami SPI i I2C,
- wyświetlacze LCD,
- układy FPGA i CPLD firm Xilinx i Altera,
- uniwersalne układy cyfrowe, w tym bramki *SingleGate*, *OneGate* z serii LVC: G00, G04, GU04, 2GU04, G07, G14, G32, G79, G86, G125, G126.

www.kamami.pl

KAMAMI

tel.: (022) 767-36-20

faks: (022) 767-36-33

List. 2. Procedura zapisu danej do rejestru układu M41T56

```

;*****
; Procedura zapisu danej do rejestru układu M41T56
; Parametry:
;   A : dana do zapisu
;   X : adres rejestru
;*****
.M41T56 WriteReg
  PUSH A           ; zapamiętanie danej na stosie
  PUSH X           ; zapamiętanie adresu rejestru na stosie
  CALL I2C_Start  ; START
  LD A, #11010000 ; adres układu M41T56
  CALL I2C_Write  ; zapis adresu układu M41T56
  POP A           ; odtworzenie adresu rejestru ze stosu
  CALL I2C_Write  ; zapis adresu rejestru wewnętrznego
  POP A           ; odtworzenie danej do zapisu ze stosu
  CALL I2C_Write  ; zapis danej do rejestru wewnętrznego
  CALL I2C_Stop   ; STOP
  RET
  
```

List. 3. Procedura odczytu danych z rejestru układu M41T56

```

;*****
; Procedura odczytu zawartości rejestru układu M41T56
; Parametry:
;   X : adres rejestru
;   Zwracana wartość :
;   A : odczytana dana
;*****
.M41T56 ReadReg
  PUSH X           ; zapamiętanie adresu rejestru na stosie
  CALL I2C_Start  ; START
  LD A, #11010000 ; adres układu M41T56
  CALL I2C_Write  ; zapis adresu układu M41T56
  POP A           ; odtworzenie adresu rejestru ze stosu
  CALL I2C_Write  ; zapis adresu rejestru wewnętrznego
  CALL I2C_Start  ; START
  LD A, #11010001 ; adres układu M41T56 + Read
  CALL I2C_Write  ; zapis adresu układu M41T56 + Read
  SCF             ; NOACK
  CALL I2C_Read   ; odczyt zawartości rejestru układu M41T56
  CALL I2C_Stop   ; STOP
  RET
  
```

List. 4. Procedura zapisu pojedynczego bajtu danych do pamięci EEPROM

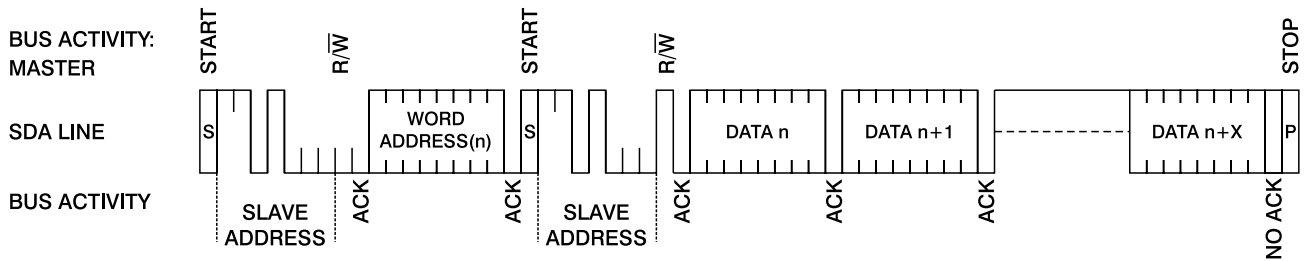
```

;*****
; Parametry :
;   m24DstAddr : adres komórki pamięci
;   A : dana do zapisu
;*****
.M24C64 WriteByte
  PUSH A           ; START
  CALL I2C_Start  ; START
  LD A, #M24WRITEADDR ; adres układu M24C64
  CALL I2C_Write  ; adres układu M24C64
  LD A, {m24DstAddr + 0} ; MSB adresu komórki pamięci
  CALL I2C_Write  ; MSB adresu komórki pamięci
  LD A, {m24DstAddr + 1} ; LSB adresu komórki pamięci
  CALL I2C_Write  ; LSB adresu komórki pamięci
  POP A           ; dana do zapisania
  CALL I2C_Write  ; dana do zapisania
  CALL I2C_Stop   ; STOP
  RET
  
```

List. 5. Procedura zapisu strony pamięci

```

;*****
; Parametry :
;   m24SrcAddr : adres bufora z danymi
;   m24DstAddr : adres, pod który mają być zapisane dane
;   A : rozmiar bufora (max. 32 bajty)
;*****
.M24C64 WriteBuffer
  LD m24Count, A ; ilość bajtów do zapisu
  CALL I2C_Start ; START
  LD A, #M24WRITEADDR ; adres układu M24C64
  CALL I2C_Write ; adres układu M24C64
  LD A, {m24DstAddr + 0} ; MSB adresu komórki pamięci
  CALL I2C_Write ; MSB adresu komórki pamięci
  LD A, {m24DstAddr + 1} ; LSB adresu komórki pamięci
  CALL I2C_Write ; LSB adresu komórki pamięci
  CLR X           ;
;
M24C64 WriteBufferLoop
  LD A, ([m24SrcAddr.w],X) ; odczyt danych z bufora
  CALL I2C_Write ; zapis danych do pamięci EEPROM
  INC X           ; X++
  CP X, m24Count ; czy to ostatni bajt?
  JRNE M24C64 WriteBufferLoop ; jeśli nie, to skok na początek pętli
  CALL I2C_Stop   ; STOP
  RET
  
```

Rys. 7. Przebiegi na magistrali I²C podczas odczytu danych z rejestrów zegara

on sygnałem NOACK. Dokonując cyklicznej próby zaadresowania układu M24C64 i sprawdzając stan bitu ACK możliwe jest stwierdzenie, czy operacja programowania pamięci dobiegła do końca. Procedurę realizującą to zadanie przedstawiono na list. 6.

Odczyt danych z pamięci EEPROM

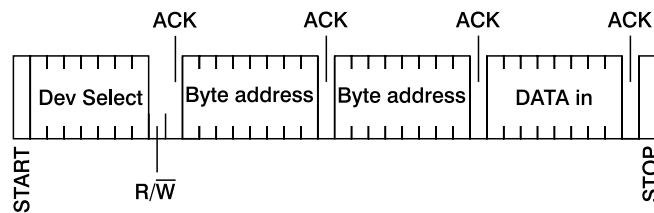
Odczyt danych z pamięci EEPROM, podobnie jak ich zapis może się odbywać na dwa sposoby: odczyt pojedynczego bajtu oraz odczyt strony. Przebiegi na magistrali podczas odczytu pojedynczego bajtu danych z pamięci przedstawiono na rys. 11. Procedurę odczytu pojedynczego bajtu spod dowolnego adresu przedstawiono na list. 7.

Przed wywołaniem procedury do zmiennej `m24SrcAddr` należy załadować adres komórki pamięci EEPROM, spod którego będą odczytywane dane. Odczytana dana zostanie umieszczona w rejestrze A.

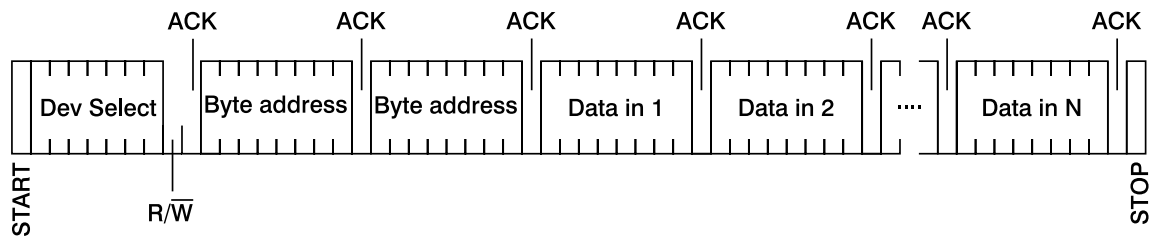
W sytuacji, gdy musimy odczytać kolejno więcej niż jeden bajt, odczytywanie pojedynczych bajtów może być bardzo uciążliwe. W związku z tym należy skorzystać z możliwości sekwencyjnego odczytu danych z pamięci M24C64. Przebiegi na magistrali podczas sekwencyjnego odczytu danych z pamięci EEPROM przedstawiono na rys. 12. Odczyt ostatniego bajtu danych musi zostać zasygnalizowany bitem NO-ACK, natomiast wszystkie poprzed-

b7	b6	b5	b4	b3	b2	b1	b0
1	0	1	0	E2	E1	E0	RW

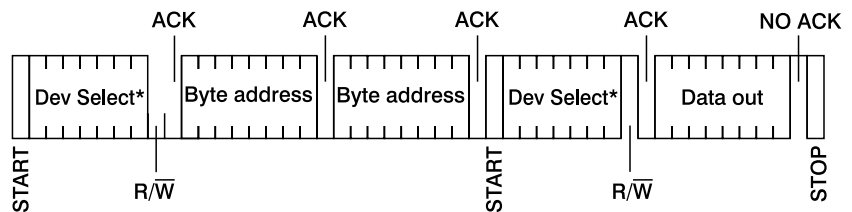
Rys. 8. Struktura bajtu adresu EEPROM-u



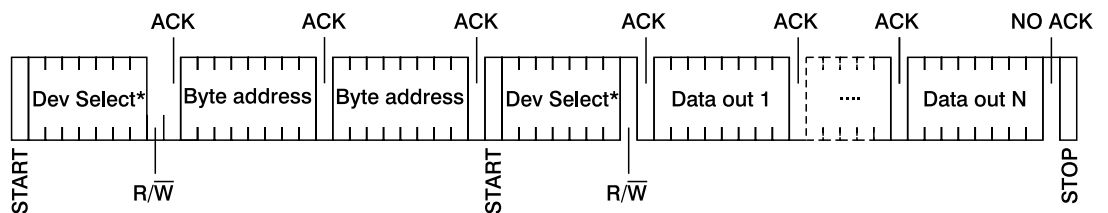
Rys. 9. Przebiegi na magistrali I²C podczas zapisu pojedynczego bajtu do EEPROM-u



Rys. 10. Przebiegi na magistrali I²C podczas zapisu strony 32 bajtów do EEPROM-u



Rys. 11. Przebiegi na magistrali I²C podczas odczytu pojedynczego bajtu z EEPROM-u



Rys. 12. Przebiegi na magistrali I²C podczas sekwencyjnego odczytu danych z EEPROM-u

nie bajty danych muszą zostać potwierdzone bitem ACK. Procedurę odczytującej dowolną ilość danych z pamięci EEPROM przedstawiono na list. 8.

Przed wywołaniem procedury w zmiennej `m24SrcAddr` należy umieścić adres, spod którego

będą odczytywane dane z pamięci EEPROM, natomiast w zmiennej `m24DstAddr` adres bufora w pamięci RAM mikrokontrolera, do którego trafią odczytane dane. W rejestrze A należy umieścić wartość określającą ilość odczytywanych bajtów danych.

Narzędzia w dobrych

cenach

dla układów CPLD/FPGA,
mikrokontrolerów AVR8,
AVR32, ARM (LPC2000,
STM32, AT91 itp.), PIC,
ST7, PSoC, MCS51

Cypress
PSoC FirstTouch
Starter Kit
120 zł

Atmel
AVR Butterfly
96 zł

STM
STM3210B
PRIMER
160 zł

XILINX
HW-CR11-SK
250 zł

STM
STM3210B-SK/KEI PROMO
400 zł
w cenie
interfejs
uLINK-EM!

ALTERA
DK-CYC11-2C20N
666 zł

ATMEL
STK500
390 zł

Podano ceny brutto

www.kamami.pl

KAMAMI

tel.: (022) 767-36-20
faks: (022) 767-36-33

List. 6. Procedura sprawdzająca gotowość układu

```
;*****  
; Procedura oczekująca na bit ACK (zakończenie operacji programowania)  
;*****  
.M24C64 WaitForACK  
CALL I2C_Start ; START  
LD A, #M24WRITEADDR ;  
CALL I2C_Write ; Adres układu M24C64  
JRC M24C64_WaitForACK ; Jeśli odebrano NOACK to skok na początek  
RET
```

List. 7. Procedura odczytu pojedynczego bajtu spod dowolnego adresu

```
;*****  
; Parametry :  
; m24SrcAddr : adres, spod którego ma zostać odczytana dana  
; Zwracana wartość :  
; A : odczytana dana  
;*****  
.M24C64 ReadByte  
CALL I2C_Start ; START  
LD A, #%10100000 ;  
CALL I2C_Write ; adres układu slave  
LD A, {m24SrcAddr + 0} ;  
CALL I2C_Write ; MSB adresu komórki pamięci  
LD A, {m24SrcAddr + 1} ;  
CALL I2C_Write ; LSB adresu komórki pamięci  
CALL I2C_Start ; START  
LD A, #%10100001 ;  
CALL I2C_Write ; adres układu slave + R  
SCF ; NOACK  
CALL I2C_Read ; odczyt bajtu danych  
CALL I2C_Stop ; STOP  
RET
```

List. 8. Procedura odczytu dowolnej ilości danych z pamięci EEPROM

```
;*****  
; Parametry  
; m24SrcAddr : adres, spod którego mają być odczytane dane  
; m24DstAddr : adres bufora z danymi  
; A : rozmiar bufora (min. 2)  
;*****  
.M24C64_ReadBuffer  
DEC A ;  
LD m24Count, A ; ilość bajtów do odczytu - 1  
CALL I2C_Start ; START  
LD A, #%10100000 ;  
CALL I2C_Write ; adres układu slave  
LD A, {m24SrcAddr + 0} ;  
CALL I2C_Write ; MSB adresu komórki pamięci  
LD A, {m24SrcAddr + 1} ;  
CALL I2C_Write ; LSB adresu komórki pamięci  
CALL I2C_Start ; START  
LD A, #%10100001 ;  
CALL I2C_Write ; adres układu slave + R  
CLR X  
M24C64_ReadBufferLoop  
RCF ; ACK  
CALL I2C_Read ; odczyt bajtu z pamięci EEPROM  
LD ([m24DstAddr.w],X),A ; zapis do bufora  
INC X ; X++  
CP X, m24Count ; czy to przedostatni bajt?  
JRNE M24C64_ReadBufferLoop ; jeśli nie to skok na początek pętli  
SCF ; NOACK  
CALL I2C_Read ; odczyt ostatniego bajtu z pamięci EEPROM  
LD ([m24DstAddr.w],X),A ; zapis do bufora  
CALL I2C_Stop ; STOP  
RET
```

Podsumowanie

Przedstawione w artykule procedury są przeznaczone co prawda dla mikrokontrolerów ST7, jednakże mogą posłużyć za wzór przy pisaniu procedur obsługi układu M41T56C64 dla innych mikrokontrolerów oraz języków programowania. W przypadku tworzenia programu w języku dostarczającym gotowe procedury obsługi magistrali I²C, jak np. Bascom, czy też kompilatory mikroBasic/microPascal/mikroC (<http://www.mikroe.com/>) wystarczy we właściwej kolejności wywołać podstawowe procedury od-

powiedzialne za transmisję sygnałów START, STOP oraz nadawanie i odbiór bajtu danych z magistrali. W przypadku tworzenia programu w asemblerze innej rodziny mikrokontrolerów należy dodatkowo opracować własne procedury obsługi magistrali I²C, bądź też skorzystać z gotowych procedur znalezionych np. w Internecie. Listingi przedstawionych w artykule procedur są zamieszczone na stronie internetowej Elektroniki Praktycznej oraz na płycie CD.
Radosław Kwiecień, EP
radoslaw.kwiecien@ep.com.pl