

Obsługa kart pamięci Flash za pomocą mikrokontrolerów, część 2

Protokół komunikacji z kartą CF

Niezależnie od wybranego trybu pracy, karty typu CF komunikują się z otoczeniem za pomocą protokołu zgodnego ze specyfikacją ATA. Oznacza to, że dostęp do zawartości pamięci jest zawsze realizowany poprzez zapis lub odczyt podstawowych rejestrów ATA. W zależności od trybu, sposób adresowania tych rejestrów wymaga nieco innej kombinacji sygnałów sterujących.

Ponieważ do komunikacji z mikrokontrolerem wybraliśmy tryb *PC Card Memory*, zostanie on najdokładniej opisany. Wymagane kombinacje sygnałów sterujących w tym trybie oraz odpowiadające im rejestry ATA zestawiono w **tab. 3**.

Zawartość pamięci Flash podzielona jest na sektory (przeważnie o długości 512 bajtów). Dostęp do danego sektora danych najłatwiej jest realizować w trybie LBA (*Logical Block Addressing*).

Do zaadresowania sektora używa się jednej 28-bitowej liczby określającej numer sektora. Wpisujemy ją do rejestrów ATA o offsetach 3, 4, 5 i 6 i ustawiamy bit 6 rejestru numeru głowicy/dysku (offset 6) dla włączenia trybu LBA. Do rejestru

W drugiej części artykułu przedstawiamy podstawy obsługi sterownika karty oraz składnię i znaczenie najważniejszych poleceń sterujących jego pracą.

Przedstawione informacje są podstawowymi i wprowadzającymi do kolejnej części artykułu, w której opiszemy programową obsługę pamięci Compact Flashy i Multimedia Card.

Ilość sektorów wpisujemy ile kolejnych sektorów chcemy odczytać lub zapisać za pomocą jednej komendy, a do rejestru komend - odpowiednią komendę (zapis/odczyt sektora). Następnie po sprawdzeniu flagi gotowości możemy rozpocząć zapis lub odczyt z rejestru danych zawartości zaadresowanego sektora. Sprawdza się to do 512-krotnego zapisania lub odczytania rejestru danych (o offsecie 0).

Jeśli wykorzystujemy 16-bitowy tryb dostępu, to dokonujemy jedynie 256 zapisów/odczytów 16-bitowych słów. Jeśli zamierzamy zapisać lub odczytać więcej niż 1 sektor danych, to przed rozpoczęciem zapisu/odczytu każdego kolejnego sektora musimy ponownie sprawdzić flagi BUSY oraz DRQ. Jeśli do rejestru *Ilość sektorów* wpisujemy 0 - oznacza to, że chcemy zapisać/odczytać 256 sektorów.

Na **rys. 6** przedstawiono algorytm dostępu do danych według standardu ATA.

Rejestry ATA

Znaczenie rejestru danych oraz rejestrów o offsetach od 2 do 6 opisano powyżej. Dla pełnego opisu rejestrów ATA należy jeszcze scharakteryzować znaczenie poszczególnych bitów rejestrów: błędów, statusu, alternatywnego statusu, rejestru kontrolnego oraz rejestru numeru głowicy/dysku - gdyż oprócz najstarszej części adresu LBA zawiera on istotne w procesie komunikacji pola bitowe.

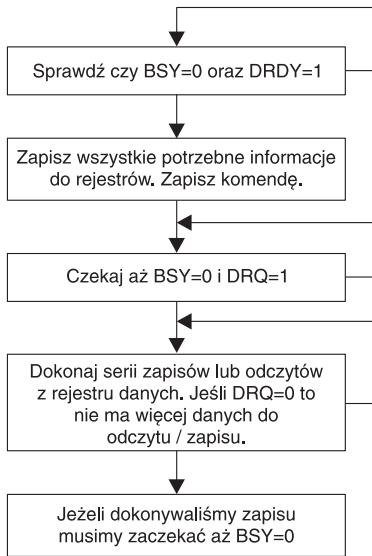
Rejestr błędów (Error Register) - offset 1, tylko do odczytu. Rejestr ten zawiera szczegółowe informacje o błędach, jeśli bit 0 rejestru statusu został ustawiony, co oznacza wystąpienie błędu. Znaczenie poszczególnych bitów tego rejestru po-

Tab. 3. Zestawienie wymaganych kombinacji sygnałów sterujących w trybie PC Card Memory oraz odpowiadające im rejestry ATA

-REG	A10	A9...A4	A3	A2	A1	A0	Offset	-OE = 0	-WE = 0
1	0	X	0	0	0	0	0	Odczyt rejestru danych	Zapis rejestru danych
1	0	X	0	0	0	1	1	Rejestr błędów	Rejestr <i>Features</i>
1	0	X	0	0	1	0	2	Ilość sektorów	Ilość sektorów
1	0	X	0	0	1	1	3	Numer sektora (LBA 7...0)	Numer sektora (LBA 7...0)
1	0	X	0	1	0	0	4	Numer cylindra [bity 0...7] (LBA 15...8)	Numer cylindra [bity 0...7] (LBA 15...8)
1	0	X	0	1	0	1	5	Numer cylindra [bity 8...15] (LBA 23...16)	Numer cylindra [bity 8...15] (LBA 23...16)
1	0	X	0	1	1	0	6	Numer głowicy/dysku (LBA 27...24)	Numer głowicy/dysku (LBA 27...24)
1	0	X	0	1	1	1	7	Rejestr statusu	Rejestr komend
1	0	X	1	0	0	0	8	Duplikat rejestru danych	Duplikat rejestru danych
1	0	X	1	0	0	1	9	Duplikat rejestru danych H	Duplikat rejestru danych H
1	0	X	1	1	0	1	D	Duplikat rejestru błędów	Duplikat rejestru <i>Features</i>
1	0	X	1	1	1	0	E	Alternatywny status	Rejestr kontrolny
1	0	X	1	1	1	1	F	Adres urządzenia	Zarezerwowany
1	1	X	X	X	X	0	8	Odczyt rejestru danych L	Zapis rejestru danych L
1	1	X	X	X	X	1	9	Odczyt rejestru danych H	Zapis rejestru danych H

Uwagi:

Rejestr o offsecie 0 może być wykorzystany w trybie 16-bitowego dostępu przy $-CE1$ i $-CE2 = 0$. W tym przypadku linie danych D0...D15 zawierają 16-bitowe dane (dwa kolejne bajty z pamięci). Jeśli $-CE1 = 0$ i $-CE2 = 1$, to mamy do czynienia z dostępem 8-bitowym i każde kolejne odwołanie do tego rejestru daje nam kolejny bajt z danego sektora danych.



Rys. 6. Algorytm dostępu do danych przy użyciu komend ATA

kazano na rys. 7. Oznaczenia na rysunku są następujące:

BBK - Bit zostaje ustawiony, jeżeli kontroler wykryje, że odwołujemy się do uszkodzonego bloku pamięci (*Bad Block*).

UNC - Bit zostaje ustawiony, jeżeli kontroler karty wykryje błąd niemożliwy do skorygowania przez wbudowane mechanizmy korekcji błędów.

IDNF - Żądany sektor nie może być odnaleziony.

ABRT - Bit zostaje ustawiony, jeśli żądana komenda została odrzucona ze względu na niegotowość karty, błędu zapisu lub próby wykonania nieprawidłowej komendy.

AMNF - Bit zostaje ustawiony, jeśli wykryto dowolny błąd.

Rejestr Statusu i Alternatywnego statusu (Status and Alt. Status Registers) - offsety 7 oraz E, tylko do odczytu. Rejestry te zwracają aktualny status karty Compact Flash. Powinien on być sprawdzany przed wysłaniem jakiegokolwiek komendy do kontrolera karty, następnie po wysłaniu tej komendy oraz pomiędzy zapisem/odczytem każdego sektora danych. Rejestr alternatywnego statusu różni się tym od podstawowego, że odczyt pierwszego nie kasuje sygnału aktywnego przerwania

D7	D6	D5	D4	D3	D2	D1	D0
BBK	UNC	0	IDNF	0	ABRT	0	AMNF

Rys. 7. Znaczenie poszczególnych bitów rejestru błędów

D7	D6	D5	D4	D3	D2	D1	D0
BUSY	RDY	DWF	DSC	DRQ	CORR	0	ERR

Rys. 8. Znaczenie poszczególnych bitów rejestrów: statusu i alternatywnego statusu

(w trybie IDE). Znaczenie poszczególnych bitów tego rejestru jest następujące (ich rozmieszczenie pokazano na rys. 8):

BUSY - Ustawiony, jeśli karta wykonuje wewnętrzne operacje i nie umożliwia w tym czasie dostępu do pozostałych rejestrów. Jeśli bit ten jest ustawiony, to pozostałe bity nie posiadają prawidłowych informacji (są nieważne).

DRDY - Zostaje ustawiony, jeśli karta jest gotowa do przyjęcia komendy. Zerowany podczas zerowania karty.

DWF - Ustawienie tego bitu oznacza błąd zapisu do karty.

DSC - Ustawiony, jeśli karta jest gotowa.

DRQ - Ustawiony, jeśli karta żąda zapisu danych do rejestru danych lub w buforze są jeszcze jakieś dane do odczytu poprzez rejestr danych.

CORR - Zostaje ustawiony, jeśli wykryto błąd danych, który został skorygowany. Ustawienie tego bitu nie przerywa transmisji danych.

O (IDX) - Zawsze ustawiony na 0.

ERR - Ustawiony w przypadku wystąpienia błędu w wyniku wykonania poprzedniej komendy. Transmisja danych zostaje przerwana. Przynajmniej wystąpienia tego błędu dodatkowo pokazuje rejestr błędów.

Rejestr Kontrolny (Device Control Register) - offset E, tylko do zapisu. Rejestr ten służy do programowego zerowania karty oraz do ustawienia trybu pracy z wykorzystaniem przerwań. Może być on zapisywany nawet w przypadku zajętości karty (bit **BUSY** w rejestrze statusu ustawiony na 1). Znaczenie poszczególnych bitów tego rejestru (rys. 9) jest następujące:

D7...D4 - Nieistotne.

D3 - Wartość tego bitu jest ignorowana przez kartę.

SW Rst - Wpisanie jedynki powoduje programowy reset karty. Wymaga programowego wyzerowania, aby zakończyć procedurę resetu i umożliwić karcie normalną pracę.

-IE_n - Jeśli jest ustawiony, karta nie zgłasza przerwań do mikro-

Tab. 4. Adresy rejestrów karty

Offset Rejestru	Adres dostępu - rys. 3	Adres dostępu - rys. 4
0	0xF00	0x8000
1	0xF10	0x8001
2	0xF20	0x8002
3	0xF30	0x8003
4	0xF40	0x8004
5	0xF50	0x8005
6	0xF60	0x8006
7	0xF70	0x8007
E	0xFE0	0x800E

Uwaga! Rysunki: 3 i 4 opublikowano w pierwszej części artykułu, w EP2/2004.

kontrolera. Najczęściej nie używa się trybu przerwań, więc należy go ustawić na 1.

D0 - Wartość tego bitu jest ignorowana przez kartę.

Rejestr głowicy/dysku (Drive/Head Register) - offset 6, do zapisu i odczytu. Rejestr ten zawiera 4 najbardziej znaczące bity adresu LBA (lub numer głowicy przy dostępie w trybie CHS), znacznik informujący kontroler o chęci dostępu do urządzenia Master lub Slave oraz znacznik wymuszający pracę w trybie LBA. Znaczenie poszczególnych bitów tego rejestru (rys. 10) jest następujące: Bity D7 i D5 - powinny być ustawione na 1.

LBA - 1 oznacza żądanie dostępu w trybie LBA.

DRV - 0 oznacza, że odwołujemy się do urządzenia Master, 1 - do Slave. Ponieważ karta CF po sieci jest skonfigurowana jako Master, powinniśmy pozostawić ten bit wyzerowany podczas komunikacji z kartą. Przewrót karty na Slave, jeśli korzystamy z trybu *PC Card Memory*, możliwe jest jedynie poprzez wpis do rejestrów konfiguracyjnych karty (dostęp przy niskim poziomie sygnału -REG), a więc stosując połączenia według rys. 3 lub rys. 4 (EP2/2004), po prostu jest niemożliwe do zrealizowania.

HS3...HS0 - 4 najbardziej znaczące bity adresu LBA lub numer głowicy w trybie CHS.

D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	1	SW Rst	+IE _n	0

Rys. 9. Znaczenie poszczególnych bitów rejestru kontrolnego

D7	D6	D5	D4	D3	D2	D1	D0
1	LBA	1	DRV	HS3	HS2	HS1	HS0

Rys. 10. Znaczenie poszczególnych bitów rejestru głowicy/dysku

0x0E - Identyfikacja napędu (Identify Drive)

Bit ->	7	6	5	4	3	2	1	0	
Command (7)	ECh								
C/D/H (6)	X	X	X	Drive	X				
Cyl High (5)					X				
Cyl Low (4)					X				
Sec Num (3)					X				
Sec Cnt (2)					X				
Feature (1)					X				

X - dane nieistotne

Drive - numer dysku - w naszym przypadku bit ten powinien mieć wartość 0

Rys. 11. Struktura kodu polecenia *identyfikacja napędu***Adresowanie rejestrów ATA**

Jeśli już znamy znaczenie poszczególnych rejestrów kontrolera ATA, musimy znać sposób ich adresowania. Zgodnie z tab. 3 musimy wygenerować na odpowiednich wyprowadzeniach karty właściwą sekwencję sygnałów sterujących. Przy wykorzystaniu interfejsu do zewnętrznej pamięci RAM zawartego w mikrokontrolerze, sprowadza się to do zapisu lub odczytu odpowiedniej komórki pamięci. W zależności od tego czy podłączyliśmy kartę według rys. 3, czy według rys. 4, poszczególne rejestry (o danym offsecie) mają adresy pokazane w tab. 4.

Jak pewnie uważni Czytelnicy zauważyli, w tab. 4 nie opisano wszystkich możliwych do zaadresowania rejestrów ATA, a jedynie 9 z nich. Owe 9 rejestrów wystarcza do typowej obsługi interfejsu ATA, a więc pozostałe rejestry możemy pominąć.

Komendy ATA

Operacje zapisu i odczytu danych z karty CF wymagają stosowania odpowiednich komend zgodnych ze specyfikacją ATA. Oznacza to przykładowo, że aby otrzymać zawartość jednego sektora pamięci, czyli 512 bajtów danych, należy najpierw zaadresować kartę jako Master, poprzez wpis do rejestru Numer głowicy/dysku czterech najbardziej znaczących bitów LBA plus wartość 0xE0, poczekać na wyzerowanie flagi BUSY i ustawienie flagi DRDY, a następnie wpisać niezbędne dane do rejestrów o offsetach od 2 do 5, po czym do rejestru komend wpisać kod komendy odczytu sektora. Następnie należy sprawdzić gotowość karty na wysłanie danych i dopiero wtedy można odczytać owe 512 bajtów poprzez wykonanie 512 odczytów *Rejestru danych* (offset 0).

Specyfikacja ATA dla kart CF zawiera dość pokaźną liczbę obsługiwanych przez nie komend ATA. Wiele z nich dotyczy trybów zabezpieczeń (*Security Commands*), oraz innych mniej ważnych funkcji lub są różnymi odmianami podstawowych komend zapisu/odczytu sektora. Ze względu na ograniczoną objętość artykułu opiszemy tylko 3 najważniejsze i niezbędne komendy.

0x0E - Identyfikacja napędu (*Identify Drive*) - **rys. 11**.

Zawartość rejestrów przy wywołaniu komendy jest następująca:

X - Dane nieistotne.

Drive - Numer dysku - w naszym przypadku bit ten powinien mieć wartość 0.

Komenda zwraca 256 słów (512 bajtów) zawierających informacje o danej karcie. Po wykonaniu komendy odczytujemy dane w identyczny sposób jak odczytujemy jeden sektor pamięci. Spośród danych najbardziej interesują nas następujące: Bajty nr 20...39 - numer seryjny karty (w ASCII).

Bajty nr 54...93 - typ (nazwa) karty.

0x20 lub 0x21 - Odczyt Sektora -ów (Read Sector (s))

Bit ->	7	6	5	4	3	2	1	0
Command (7)	20h lub 21h							
C/D/H (6)	1	LBA	1	Drive	Head (LBA 27-24)			
Cyl High (5)					Cylinder High (LBA 23-16)			
Cyl Low (4)					Cylinder Low (LBA 15-8)			
Sec Num (3)					Sector Number (LBA 7-0)			
Sec Cnt (2)					Sector Count			
Feature (1)					X			

X - dane nieistotne

LBA = 1

Drive = 0

LBA[27..0] - adres sektora, od którego rozpoczynamy transmisję

Sector Count - ilość sektorów do odczytu (0 = 256, 1 = 1, 2 = 2 itd.)

Rys. 12. Struktura kodu polecenia *odczyt sektora/ów*

Bajty nr 120...124 - liczba dostępnych sektorów w trybie LBA, czyli zarazem pojemność danej karty - to najważniejsza dla nas informacja, którą zwraca komenda identyfikacji.

0x20 lub 0x21 - Odczyt Sektora/ów (*Read Sector(s)*) - **rys. 12**.

Zakładając, że adresujemy kartę w trybie LBA, zawartość rejestrów przy wywołaniu komendy powinna wyglądać tak:

X - Dane nieistotne.

LBA = 1.

Drive = 0.

LBA[27...0] - adres sektora, od którego rozpoczynamy transmisję.

Sector Count - liczba sektorów do odczytu (0 = 256, 1 = 1, 2 = 2 itd.).

Jest to polecenie najbardziej dla nas interesujące. Pozwala na odczyt maksymalnie 256 sektorów, jeśli wartość rejestru 2 wynosi zero. Rejestry 3...6 zawierają adres początkowego sektora. Jeśli bit LBA = 1, wtedy stosujemy adresowanie LBA, a gdy LBA=0, to CHS. Po wydaniu komendy trzeba poczekać, aż kontroler karty wyzeruje flagę BUSY i ustawi DRQ, a następnie odczytać 512 bajtów danych z *Rejestru danych*. Komenda posiada dwie odmiany - wersja z powtórzeniami (komenda 0x20) nakazuje kontrolerowi w przypadku natrafienia na uszkodzony sektor próbować odczytać go ponownie. Liczba prób jest nieokreślona standardem i specyficzna dla każdego typu karty. Mechanizm ten jest niewidoczny dla użytkownika, a informacja o wystąpieniu problemu pojawia się jedynie jako ustawienie bitu CORR w rejestrze statusu - o ile kontrolerowi udało się skorygować odczytywane dane. W przeciwnym przypadku kontroler ustawia flagę ERR i przerywa trans-

0x30 lub 0x31 - Zapis Sektora -ów (Write Sector (s))

Bit ->	7	6	5	4	3	2	1	0
Command (7)	30h lub 31h							
C/D/H (6)	1	LBA	1	Drive	Head (LBA 27-24)			
Cyl High (5)	Cylinder High (LBA 23-16)							
Cyl Low (4)	Cylinder Low (LBA 15-8)							
Sec Num (3)	Sector Number (LBA 7-0)							
Sec Cnt (2)	Sector Count							
Feature (1)	X							

Rys. 13. Struktura kodu polecenia zapis sektora/ów

misję, jeśli próbowaliśmy odczytać więcej niż 1 sektor.

0x30 lub 0x31 - Zapis sektora/ów (Write Sector(s)) - **rys. 13.**

Zawartość rejestrów przy wywołaniu komendy jest identyczna jak dla komendy odczytu, z wyłączeniem kodu komendy.

Zasada działania jest identyczna jak w przypadku komendy *Read Sector*, ale w tym przypadku dane są zapisywane do karty. Po wysłaniu zawartości jednego sektora musimy poczekać, aż kontroler karty wyzeruje flagę BUSY, po czym jeś-

li chcemy zapisać więcej niż 1 sektor (*Sector Count* różny od 1), to możemy wysłać do karty zawartość kolejnego sektora, czyli kolejne 512 bajtów danych do *Rejestru danych* (offset 0).

Programowe zerowanie sterownika karty CF

Po załączeniu zasilania karty, przed rozpoczęciem komunikacji z kartą lub w przypadku wystąpienia błędów komunikacji należy wykonać programowe zerowanie sterownika karty. W praktyce polega ono na:

- Wpisaniu do rejestru kontrolnego (offset E) wartości 0x06.
- Odczekaniu około 10 μ s.
- Wpisaniu do rejestru kontrolnego wartości 0x02.
- Odczekaniu około 10 μ s.
- Sukcesywnym odczytywaniu zawartości rejestru Statusu do czasu aż kontroler wyzeruje flagę BUSY i ustawi flagę DRDY. Maksymalny czas odpowiedzi na reset może wynieść około 500 ms.
- Kontrolnego odczytu rejestru *Numer sektora* (offset 3), który po prawidłowym resetie karty przyjmuje wartość 1.

Jeśli procedura ta zakończy się sukcesem, mamy pewność, że karta jest gotowa do przyjęcia wybranej przez nas komendy.

Wszystkie podane dotychczas informacje powinny pozwolić na napisanie konkretnych procedur komunikacji z kartą CF, więc w kolejnej części kursu przedstawię sposób implementacji komend zapisu i odczytu sektora dla kart typu Compact Flash, oparty o procedury w języku „C“.

Romulad Biały