

# Obsługa kart pamięci Flash za pomocą mikrokontrolerów, część 6

## Karty MultiMedia Card (MMC)

### Potwierdzenia

Karty MMC w trybie SPI wysyłają dwa rodzaje potwierdzeń. Pierwszy z nich, to potwierdzenia odsyłane po przyjęciu komendy. Występuje ono w 3 wariantach oznaczonych jako R1, R1b i R2. Potwierdzenie R1 jest podstawowym rodzajem potwierdzeń wysyłanych po większości komend. R1b jest to potwierdzenie typu R1 uzupełnione o sygnał BUSY. Potwierdzenie R2 jest wysyłane w odpowiedzi na komendę żądania statusu karty i ono jest właśnie dwubajtowym statusem karty. Pierwszy bajt potwierdzenia R2 jest taki sam jak w potwierdzeniu R1. Dokładny format potwierdzeń po otrzymaniu komendy pokazano na rys. 7 i 8. Zgodnie ze specyfikacją MMC nie muszą być one wysyłane natychmiast po otrzymaniu komendy. Pomiędzy jej wysłaniem, a zwróceniem potwierdzenia należy wysłać na linie CLK od 8 do 64 impulsów zegarowych co oznacza, że musimy odebrać od 1 do 8 bajtów z magistrali SPI. Na szczęście najstarszy bit w bajcie potwierdzenia ma zawsze wartość „0”, a w trakcie oczekiwania na potwierdzenie karta zwraca wartość 0xFF, więc po prostu w pętli odbieramy dane z karty do czasu aż najstarszy bit odebranego bajtu będzie równy 0, co oznacza że właśnie odebraliśmy bajt potwierdzenia. W trybie SPI karta zawsze musi odebrać i potwierdzić komendę, nawet jeśli jest w trybie BUSY.

Drugim rodzajem potwierdzeń jest potwierdzenie *Data Response*, wysyłane przez kartę po przesłaniu do niej każdego bloku danych, który ma być zapisany na karcie. Jest to jednobajtowe potwierdzenie wysyłane natychmiast po otrzymaniu przez kartę kompletnego bloku danych, a jego format pokazano na rys. 9.

### Sygnal BUSY czyli zajętość karty

Po każdej operacji zapisu lub kasowania danych, wewnętrzny kontroler karty rozpoczyna wewnętrzny proces zapisywania danych do pamięci Flash. W tym czasie karta nie może wykonywać żadnych nowych komend. Zostaje on

*W przedostatniej części cyklu autor przedstawia ostatnie zagadnienia teoretyczne, których poznanie jest niezbędne do prawidłowego posługiwania się kartami pamięci MMC.*

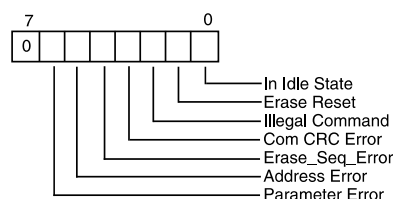
wystawiony bezpośrednio po wysłaniu po potwierdzeń typu R1b oraz *Data Response*. Sygnal BUSY to permanentne ściągnięcie linii *DataOut* do masy, odczytywane przez mikrokontroler jako bajt o wartości 0x00. W przypadku zajętości karty można zatrzymać sygnał *CLK* a nawet zdezaktywować sygnał *CS*, co spowoduje przejście linii *DataOut* w stan wysokiej impedancji. Karta będzie kontynuowała wewnętrzny proces zapisu danych. Po ponownym wybraniu karty niskim poziomem na linii *CS*, jeśli karta jest nadal w stanie BUSY, to pojawi się on na linii *DataOut* po otrzymaniu pierwszego impulsu na linii *CLK*. Najprostszym sposobem sprawdzania zajętości karty jest odczytywanie jej w pętli to czasu otrzymania bajtu o wartości różnej od 0. Jednakże nie można robić tego w nieskończoność, ponieważ fizyczne zwarście linii *DataOut* do masy spowoduje, że program utknie na oczekiwaniu na koniec zajętości karty. Specyfikacja MMC podaje, że po 10-krotnym przekroczeniu wartości typowego czasu zapisu można uznać że nastąpił jakiś nieoczekiwany błąd i należy podjąć próbę odzyskania kontroli nad kartą poprzez próbę ponownej inicjalizacji karty, lub chwilowego odłączenia zasilania karty (jeśli przewidziano taką możliwość).

### Data Error Token

W przypadku wystąpienia błędu podczas odczytu karty, zamiast bloku danych, karta zwróci nam

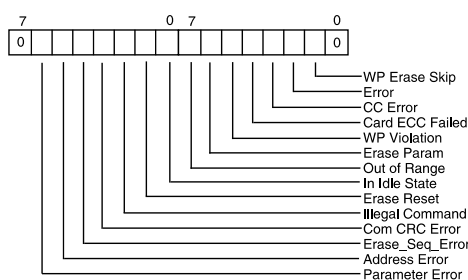


jeden bajt zwany *Data Error Token*. Oznacza to, że zamiast spodziewanego bajtu o wartości 0xFE oznaczającego początek bloku danych (bajt startu) otrzymamy bajt którego 4 najstarsze bity są równe zeru, a pozostałe zawierają przyczynę błędu. Strukturę tego bajtu pokazano na rys. 10, a znaczenie poszczególnych bitów jest takie samo jak przy opisie potwierdzenia typu R2.



*In Idle State* - Karta jest w trybie IDLE, czyli w trakcie inicjalizacji.  
*Erase Reset* - Przerwano sekwencję kasowania.  
*Illegal Command* - Błędna lub nieobsługiwana komenda (w trybie SPI).  
*Com CRC Error* - Błąd CRC w ostatnio wysłanej komendzie.  
*Erase Seq Error* - Błędna sekwencja kasowania.  
*Address Error* - Podano błędny adres, co spowodowało przekroczenie granice bloku.

Rys. 7. Potwierdzenie typu R1



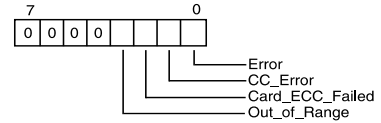
*Parameter Error* - Zły argument dla danej karty.  
*Out of Range* - Podczas operacji przekroczono granicę fizycznego bloku karty.  
*Erase Param* - Złe wybrano sektory lub grupy do kasowania.  
*WP Violation* - Próba zapisu zabezpieczonego bloku.  
*Card ECC Failed* - Próba wewnętrznej korekcji błędów nie powiodła się.  
*CC Error* - Błąd kontrolera karty.  
*Error* - Nieznany błąd.  
*WP Erase Skip* - Podczas sekwencji kasowania opuszczono część bloków ze względu na zabezpieczenie przed kasowaniem.

Rys. 8. Potwierdzenie typu R2

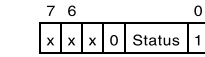
### Inicjalizacja karty

Przed rozpoczęciem jakiegokolwiek komunikacji z kartą, po załączeniu zasilania karty musi ona otrzymać co najmniej 74 cykle zegarowe na linii CLK. Oznacza to wysłanie co najmniej 8 bajtów o wartości 0xFF zanim rozpoczniemy właściwą procedurę inicjalizacji karty. Opisane 8 bajtów powinno być wysłane przy wysokim stanie na linii CS. W tym momencie interfejs karty zaczyna pracować w trybie MMC. Aby przełączyć się w tryb SPI należy wysłać do karty komendę CMD0 przy niskim stanie sygnału CS. W tym momencie karta przełącza się w tryb SPI i potwierdza ten fakt odpowiedzią R1 zgodną z trybem SPI, oraz pozostaje w stanie IDLE. Ponieważ komenda CMD0

jest wysyłana do karty w czasie gdy jest ona w trybie MMC, w którym to trybie karta ma standardowo załączony tryb sprawdzania sumy CRC, musi ona zostać wysłana wraz z prawidłową sumą kontrolną dla tej komendy. Na szczęście jest to komenda statyczna i zawsze posiada takie samo CRC, więc nie ma potrzeby jego wyliczania. Kompletna, 6-bajtowa komenda przełączająca kartę w tryb SPI wygląda tak: 0x40, 0x00, 0x00, 0x00, 0x00, 0x95. Po przełączeniu karty w tryb SPI sprawdzanie sumy kontrolnej zostaje automatycznie wyłączone, więc pisząc procedurę wysyłającą komendy do karty, można jako bajt CRC wysyłać zawsze wartość 0x95, która będzie prawidłowym CRC dla CMD0 w momencie inicjalizacji karty i będzie ignorowany przy pozostałych komendach (o ile nie zamierzamy włączyć obsługi CRC). Jedyną dozwoloną komendą w stanie IDLE, w którym to karta pozostaje po wykonaniu CMD0, jest komenda CMD1. Mikrokontroler powinien sukcesywnie wysyłać tą komendę do czasu kiedy w otrzymanym bajcie odpowiedzi bit 0 oznaczający stan IDLE nie zostanie wyzerowany. W tym momencie karta zakończyła procedurę wewnętrznego zero-



Rys. 9. Potwierdzenie Data Response



Status:  
010 – Dane zaakceptowane.  
101 – Dane odrzucone ze względu na błąd CRC.

Rys. 10. Data Error Token

**Tab. 5. Definicje czasów – jednostką jest 8 cykli zegarowych**

Symbol	Minimum	Maksimum
$N_{CS}$	0	–
$N_{CR}$	1	8
$N_{RC}$	1	–
$N_{AC}$	1	$10 \cdot (TAAC + NSAC)$
$N_{WR}$	1	–
$N_{EC}$	0	–
$N_{DS}$	0	–

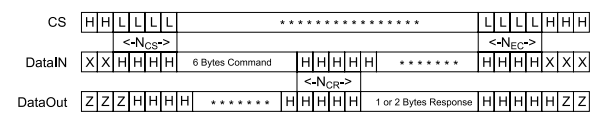
wania i jest gotowa na przyjęcie następnej komendy. W tym momencie proces inicjalizacji dobiegł końca i karta jest gotowa do pracy.

### Zależności czasowe

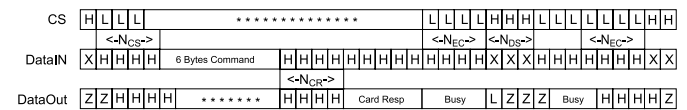
Na koniec tej części artykułu przedstawiam zależności czasowe występujące podczas komunikacji z kartą MMC z wykorzystaniem protokołu SPI. Pochodzą one z specyfikacji dostarczonej przez firmę SanDisk, którą mogą czytelnicy znaleźć na CD-EP8/2004B.

W kolejnej – ostatniej – części kursu zajmiemy się praktycznymi przykładami procedur obsługi kart MMC napisanymi w języku C dla mikrokontrolerów AVR.

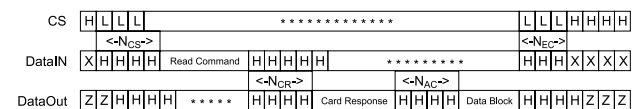
**Romuald Biały**



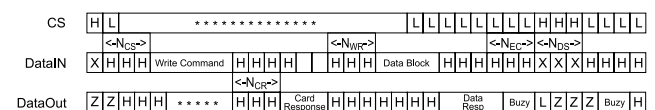
Odpowiedź karty na komendę – karta jest gotowa.



Odpowiedź karty na komendę – karta jest zajęta.



Odczyt danych.



Zapis danych.