

Obsługa kart pamięci Flash za pomocą mikrokontrolerów, część 5 Karty MultiMedia Card (MMC)

Protokół komunikacji z kartą MMC

Komunikacja pomiędzy mikroprocesorem a kartą w trybie SPI realizowana jest w czasie, kiedy sygnał selekcji CS przyjmuje stan niski. Po wystawieniu „0” linii CS należy kolejno wykonać następujące operacje:

1. Wysłać do karty 6-bajtową komendę, która składa się z bajtu numeru, 4 bajtów argumentu komendy oraz bajtu sumy CRC.

2. Odebrać z karty potwierdzenie wykonania komendy, które w zależności od komendy jest jedno- lub dwubajtowe.

3. Wysłać lub odebrać blok danych. W zależności od komendy, blok danych może w ogóle nie występować, a jeśli występuje, to może mieć długość od 4 do 512 bajtów. W jego skład wchodzi bajt startu o stałej wartości 0xFE, następnie występuje blok od 1 do 512 bajtów właściwych danych oraz dwa bajty sumy kontrolnej CRC.

Kontynuujemy prezentację sposobu obsługi kart pamięciowych MMC. W tej części kursu przedstawiamy rozkazy sterujące pracą sterownika karty.

W przypadku gdy komenda odczytu danych nie powiedzie się i karta nie może zwrócić właściwych danych, to zamiast bloku danych wysyłany jest jednobajtowy kod błędu zwany *Data Error Token*.

4. W przypadku zapisu do karty, dodatkowo należy odebrać z karty potwierdzenie przyjęcia bloku danych oraz odczekać na zakończenie operacji zapisu poprzez sprawdzenie stanu BUSY.

5. Zmienić stan linii CS na wysoki, po czym wysłać do karty 8 impulsów zegarowych, co oznacza wysłanie jednego bajtu danych o wartości 0xFF.

Na tym kończy się procedura obsługi pojedynczej komendy do karty MMC z wykorzystaniem trybu SPI.



Standardowo karta MMC pracująca w trybie SPI ma wyłączone sprawdzanie poprawności CRC zarówno dla komend, jak i dla bloków danych, co upraszcza procedury obsługi. Tak więc poza jednym przypadkiem, o którym napiszę przy okazji opisu inicjalizacji karty, nie musimy się przejmować wartościami sum CRC, lecz muszą one zostać wysłane (lub odebrane) do/z karty. Jeśli zależy nam na

Tab. 4. Spis komend MMC dostępnych w trybie SPI

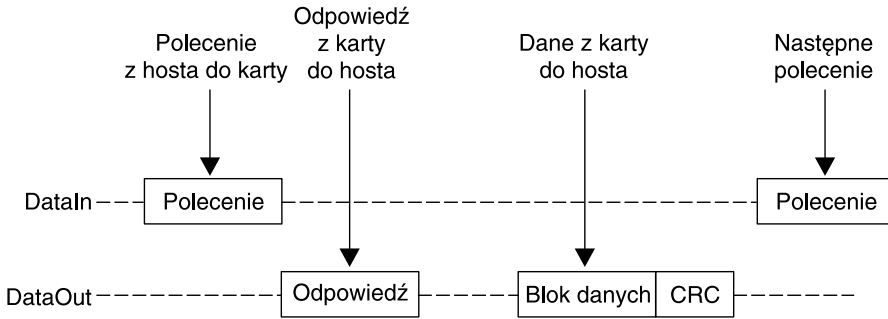
Numer CMD	Nazwa komendy	Argument	Format odpowiedzi	Opis
CMD0	GO_IDLE_STATE	Brak	R1	Zerowanie karty MMC
CMD1	SEND_OP_COND	Brak	R1	Uruchomienie procedury inicjalizacji karty
CMD9	SEND_CSD	Brak	R1	Odczyt rejestru CSD
CMD10	SEND_CID	Brak	R1	Odczyt rejestru CID
CMD13	SEND_STATUS	Brak	R2	Odczyt statusu karty
CMD16	SET_BLOCKLEN	Długość bloku	R1	Ustawienie długości bloku danych (w bajtach) dla komend zapisu i odczytu (**)
CMD17	READ_SINGLE_BLOCK	Adres danych	R1	Odczyt bloku danych z karty, o długości ustawionej komendą SET_BLOCKLEN (***)
CMD24	WRITE_BLOCK	Adres danych	R1 + DR*	Zapis 512-bajtowego bloku danych na kartę (***)
CMD27	PROGRAM_CSD	Brak	R1 + DR*	Zapis możliwych do zmiany bitów rejestru CSD
CMD28	SET_WRITE_PROT	Adres danych	R1b	Włącza zabezpieczenie przed zapisem dla danej grupy bloków
CMD29	CLR_WRITE_PROT	Adres danych	R1	Wyłącza zabezpieczenie przed zapisem dla danej grupy bloków
CMD30	SEND_WRITE_PROT	Adres danych	R1	Odczyt statusu zabezpieczeń przed zapisem (****)
CMD32	TAG_SECTOR_START	Adres danych	R1	Ustawia adres pierwszego sektora dla grupy do skasowania
CMD33	TAG_SECTOR_END	Adres danych	R1	Ustawia adres ostatniego sektora dla grupy do skasowania, lub adres pojedynczego sektora do skasowania
CMD34	UNTAG_SECTOR	Adres danych	R1	Usuwa wybrany sektor z grupy do skasowania
CMD35	TAG_ERASE_GROUP_START	Adres danych	R1	Ustawia adres pierwszej grupy do skasowania
CMD36	TAG_ERASE_GROUP_END	Adres danych	R1	Ustawia adres ostatniej grupy do skasowania
CMD37	UNTAG_ERASE_GROUP	Adres danych	R1	Usuwa jedną, wcześniej wybraną grupę z zakresu kasowania
CMD38	ERASE	Brak	R1b	Kasowanie wszystkich wcześniej wybranych sektorów
CMD59	CRC_ON_OFF	1 lub 0 na najmłodszym bicie argumentu	R1	Włączenie lub wyłączenie sprawdzania sum CRC (1 – załączenie, 0 – wyłączenie)

* DR - dodatkowe potwierdzenie po wysłaniu bloku danych plus status BUSY.

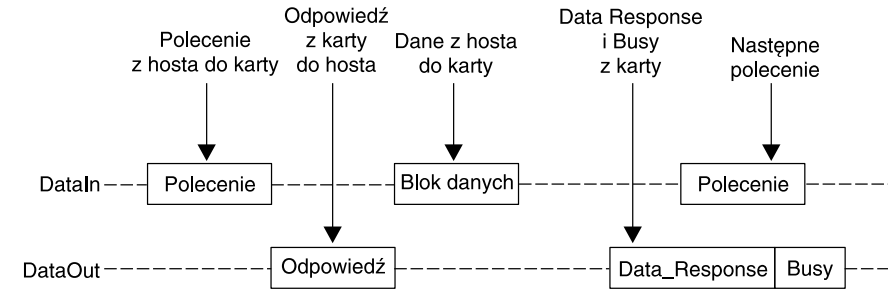
** Podczas zapisu, dopuszcza się jedynie bloki o długości 512 bajtów, ustawienie innej długości spowoduje błąd zapisu. Przy odczycie można ustawić długość bloku w zakresie od 1 do 512 bajtów.

*** Adres danych i długość bloku muszą być tak dobrane, aby podczas tych operacji nie zostały przekroczone granice pojedynczego fizycznego bloku na karcie.

**** Odczyt 32 bitów (4 bajtów) reprezentujących 32 grupy bloków, które można zabezpieczyć przed zapisem, plus dwa bajty CRC.



Rys. 3. Odczyt danych z karty MMC



Rys. 4. Zapis danych do karty

pewności transmisji, możemy za pomocą specjalnej komendy załączyć sprawdzanie sumy kontrolnej, ale będzie to wymagało uzupełnienia oprogramowania o procedury liczenia i kontroli CRC.

Na rys. 3, 4 i 5 pokazano szczegółowy przebieg operacji odczytu, zapisu oraz komend bez bloku danych.

Komendy

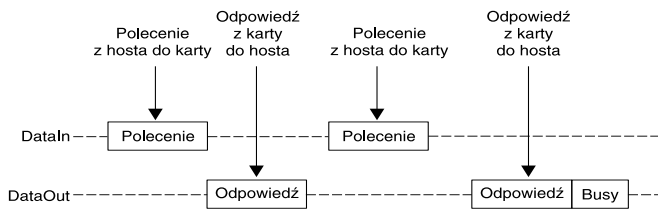
Karta MMC komunikuje się z *hostem*, wykorzystując szereg komend zdefiniowanych standardem MMC. Nie wszystkie komendy są dostępne w trybie MMC. Na rys. 6 pokazano format każdej z komend, a w tab. 4 szczegółowy spis komend dostępnych w trybie SPI wraz z ich opisem. Każda komenda składa się z 6 bajtów. Pierwszy bajt komendy określa jej numer oraz musi mieć szósty bit ustawiony na „1”, a więc na przykład dla komendy

CMD13 pierwszy bajt komendy wynosi $13+64 = 77$ (0x4D). Bajty od 2 do 5 zawierają argument dla komendy. Jeśli komenda nie posiada dodatkowych argumentów, jako argument wysyłamy 4 bajty o wartości 0x00. W przypadku komend

zapisu i odczytu bloku, gdy argumentem jest adres danych, podajemy adres pierwszego bajtu do odczytu/zapisu – a nie numer bloku! Oznacza to, że jeśli chcemy operować numerami 512 bajtowych sektorów – tak jak w przypadku kart CF – to numer sektora należy pomnożyć przez 512 i dopiero wtedy wpisać jako argument komendy zapisu czy odczytu. Szósty bajt komendy zawiera sumę CRC7 plus bit o wartości 1 na najmłodszej pozycji. Praktycznie, nie wykorzystując sprawdzania sum CRC, zawsze wysyłamy w tym miejscu wartość 0x95. A dlaczego – napiszę przy okazji opisu inicjalizacji karty.

Jak widać, wiele komend dotyczy procesu kasowania bloków. Ma on na celu przyspieszenie zapisu do karty, ponieważ można za jednym „zamachem” skasować wiele bloków jednocześnie, a następnie dopiero rozpocząć ich zapis, który w tym przypadku jest szybszy. Ze względu na to, że wcześniejsze kasowanie bloków nie jest konieczne oraz na skomplikowane procedury kasowania, nie będę ich tu opisywał. W większości przypadków wystarczy fakt, że jeśli dany blok nie został wcześniej skasowany i wywołamy komendę zapisu bloku, to najpierw zostanie on automatycznie skasowany, a następnie zapisany nową zawartością.

Romuald Biały



Rys. 5. Komendy niezawierające dodatkowych danych

Bajt 1		Bajty 2...5				Bajt 6			
7	6	5	0	31	0	7	0		
0	1	Polecenie				Argument polecenia		CRC	1

Rys. 6. Format komend MMC w trybie SPI