# Czytnik-programator kart chipowych, część 2

## kit AVT-468

*Kończymy prezentację konstrukcji czytnika - programatora kart chipowych. W tej części artykułu przedstawimy listę rozkazów wykorzytywanych do sterowania pracą karty chipowej oraz sposób wymiany informacji pomiędzy mikrokontrolerem obsługującym kartę, a programem terminalowym. Skrótowo przedstawimy także bardzo efektowną, przykładową aplikację czytnika.*


### Rozkazy sterujące pracą karty X76F640

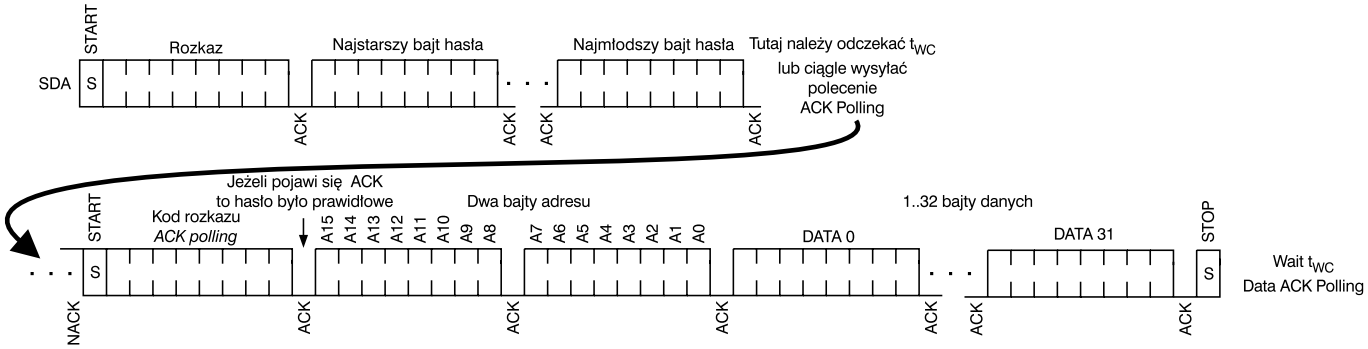
Jak wspomniano w pierwszej części artykułu transfer danych do i z karty odbywa się poprzez dwuprzewodową magistralę, która jest funkcjonalnie bardzo podobna do znanego standardu I2C. W identyczny sposób generowane są przez Mastera warunki *Start* i *Stop*, w taki sam sposób przebiega transmisja każdego bitu, nieco inaczej za to odbierane jest potwierdzenie *ACK*.

Pomocny w wyjaśnieniu problemu będzie **rys. 11**. Wykonanie jakiegokolwiek polecenia przez wewnętrzny automat sterujący karty chipowej jest uwarunkowane podaniem przez użytkownika uprawnionego hasła o długości 64 bitów. Hasło jest wysyłane zawsze po bajcie polecenia (rys. 11). Każdy wpis hasła do karty powoduje uruchomienie cyklu zapisu matrycy EEPROM, który trwa ok. 5 ms. Dopiero po tym czasie karta jest w stanie odpowiedzieć użytkownikowi, czy podane hasło jest prawidłowe, czy też nie. Zastosowanie tak długiego czasu odpowiedzi zostało podyktowane chęcią utrudnienia życia potencjalnym włamywaczom. Pojawienie się sygnału ACK oznacza, że hasło było prawidłowe. Jeżeli w ciągu 10 ms karta nie wygeneruje potwierdzenia należy uznać, że hasło było błędne.

Podanie błędnego hasła powoduje automatyczne zwiększenie stanu licznika błędnych prób! Jeżeli nieuprawnione próby będą ponawiane, to po ósmej karta zostaje zablokowana!

Sterowanie pracą karty umożliwia 12 rozkazów o kodach podanych w **tab. 1**. Pokrótce je omówimy:

- ***READ ARRAY 0/1***. Polecenia odczytu zawartości komórki wybranej matrycy pamięciowej o adresie podanym w dwóch kolejnych (po haśle) bajtach. Karta po odebraniu tego polecenia wysyła w odpowiedzi wskazany bajt danych. Możliwe jest także losowe odczytanie dowolnej z 256 kolejnych komórek pamięci poprzez wysłanie do karty mniej znaczącej części adre-




Rys. 11. Format ramki danych podczas zapisu sektora.

su. Adres ten musi zostać poprzedzony generowanym przez *Mastera* warunkiem *START*. W przypadku pominięcia warunku *START* kolejne takty zegarowe wysyłane do karty spowodują wysyłanie przez nią bajtów o kolejnych adresach. Tak więc możliwy jest odczyt zawartości kilku komórek pamięci po jednokrotnym podaniu adresu, bez konieczności dodatkowego adresowania każdego odczytu.

- ***SECTOR WRITE 0/1***. Rozkaz umożliwiający zapis informacji do matrycy pamięciowej pod wybrany adres. Podobnie jak w przypadku polecenia odczytu, jako następne musi zostać przesłane do karty hasło dostępowe. Możliwy jest jednoczesny wpis 32 bajtów danych, które zapisywane są w matrycy EEPROM w jednym cyklu programowania. Każda transmisja bloku danych wpisywanych do pamięci musi być zakończona znakiem *STOP*.
- ***CHANGE READ 0/1 PASSWORD***. Polecenie umożliwiające zmianę dotychczasowego hasła zabezpieczającego obydwie matryce kart przed odczytem. Zmiana każdego z tych haseł wymaga podania poprzedniego hasła, co zapobiega możliwości nieuprawnionego dostępu do zawartości karty.
- ***CHANGE WRITE 0/1 PASSWORD***. Polecenie podobne do poprzedniego, z tą różnicą, że dotyczy hasła zabezpieczającego kartę przed zapisem.
- ***CHANGE RESET PASSWORD***. Rozkaz umożliwiający zmianę dotychczasowego hasła dostępu do polecenia zerowania zawartości karty.
- ***RESET PASSWORD***. Rozkaz powodujący wyzerowanie zawartości rejestrów haseł oraz obydwu matryc pamięciowych *Array0* i *Array1*. Wysłanie tego polecenia do karty jest najprostszym sposobem jej szybkiego, całkowitego wyzerowania.
- ***RESET DEVICE***. Rozkaz umożliwiający uruchomienie karty po zablokowaniu jej przez licznik nieuprawnionych prób dostępu. Przy pomocy tego polecenia można także wyzerować ten licznik, przed przekroczeniem dopuszczalnego limitu prób (8).
- ***ACK POLLING***. Polecenie kończące procedurę zapisu do pamięci hasła (rys. 11). Można je także wykorzystać do kontroli aktualnego stanu karty (czy został zakończony proces zapisu matrycy EEPROM).

Kody instrukcji, które nie zostały wymienione w tab. 1 są nielegalne i nie powinny być stosowane.

## Sterowanie czytnika - programatora

Wymienione wcześniej rozkazy umożliwiają bezpośrednią komunikację z kartą, stanowiąc najniższą warstwę komunikacji.

Zastosowanie w programatorze mikrokontrolera zwalnia użytkownika w znacznym stopniu z konieczności poznania szczegółów dotyczących transmisji danych oraz wszelkich niuansów wynikających ze specyfikacji producenta. Z tego też powodu powstał język nieco wyższego poziomu, przy pomocy którego bez trudu można kartę zaprogramować lub odczytać jej zawartość, wykorzystując standardowe programy terminalowe.

Składnia tego języka jest bardzo prosta - szczegóły przedstawiamy poniżej.

### Odczyt matryc pamięciowych *Array0* i *Array1*

*Składnia polecenia:*

Rd***a***:***xxxxxxxxxxxxxxxx***:***AAAA***

***a*** - numer matrycy 0 lub 1;

***x*** - znaki szesnastkowe 0..F składające się na hasło *READ0/1*;

***A*** - znaki szesnastkowe 0..F składające się na adres początkowy (zakres 0000..1FFFh).

W odpowiedzi na takie polecenie sterownik odeśle (jeżeli hasło dostępu było poprawne!) zawartość wybranej komórki pamięci.

### Odczyt zawartości bloków matryc pamięciowych *Array0* i *Array1*

*Składnia polecenia:*

Ra***a***:***xxxxxxxxxxxxxxxx***:***AAAA***

***a*** - numer matrycy 0 lub 1;

***x*** - znaki szesnastkowe 0..F składające się na hasło *READ0/1*;

***A*** - znaki szesnastkowe 0..F składające się na adres początkowy (zakres 0000..1FFFh).

W odpowiedzi na takie polecenie sterownik odeśle (jeżeli hasło dostępu było poprawne!) 256 kolejnych bajtów wybranej matrycy pamięciowej, począwszy od podanego adresu. Jeżeli adres początkowy ***AAAA*** będzie większy od 1F00h wysłane zostaną także kolejne bajty począwszy od adresu 0.

**Tab. 1.**

| Bajt polecenia | Hasło (64 bity) | Starszy bajt dodatkowy | Młodszy bajt dodatkowy | Opis |
|---|---|---|---|---|
| 10000000 | READ0 | Adres MSB | Adres LSB | Odczyt matrycy Array0 |
| 10001000 | READ1 | Adres MSB | Adres LSB | Odczyt matrycy Array1 |
| 10010000 | WRITE0 | Adres MSB | Adres LSB | Zapis sektora w matrycy Array0 |
| 10011000 | WRITE1 | Adres MSB | Adres LSB | Zapis sektora w matrycy Array1 |
| 10100000 | READ0 | 00000000 | 00000000 | Zmiana hasła READ0 |
| 10101000 | READ1 | 00000000 | 00000000 | Zmiana hasła READ1 |
| 10110000 | WRITE0 | 00000000 | 00000000 | Zmiana hasła WRITE0 |
| 10111000 | WRITE1 | 00000000 | 00000000 | Zmiana hasła WRITE1 |
| 11000000 | RESET | 00000000 | 00000000 | Zmiana hasła RESET |
| 11100000 | RESET | brak | brak | Zerowanie rejestrów haseł oraz obydwu matryc pamięciowych |
| 11101000 | RESET | brak | brak | Zerowanie licznika błędnych prób dostępu i ew. zawartości matryc oraz rejestrów haseł |
| 11110000 | brak | brak | brak | Sygnalizacja końca wpisu hasła. Polecenie wykorzystywane przez procedurę ACK Polling |

## Zapis sektora matryc pamięciowych *Array0* i *Array1*

*Składnia polecenia:*

Wr***a***:***xxxxxxxxxxxxxxxx***:***AAAA***

***a*** - numer matrycy 0 lub 1;

***x*** - znaki szesnastkowe 0..F składające się na hasło *WRITE0/1*;

***A*** - znaki szesnastkowe 0..F składające się na adres początkowy (zakres 0000..1FFFh).

Wysłanie takiego ciągu znaków do programatora przełącza go w tryb oczekiwania na dane - kolejne 32 bajty (jeżeli nie będzie to ciąg znaków *RST*) zostaną wpisane po zadany adres.

## Anulowanie ostatniego polecenia

*Składnia polecenia:*

RST

Wysłanie polecenia *RST* powoduje natychmiastowy powrót kontrolera programatora do stanu oczekiwania (logicznego wyzerowania). Dzięki temu można np. anulować ostatnio wysłane polecenie zapisu matrycy pamięciowej.

Polecenie to nie ma żadnego wpływu na kartę.

## Zmiana haseł zapisu

*Składnia polecenia:*

Cw***a***:***xxxxxxxxxxxxxxxx***:***yyyyyyyyyyyyyyyy***

***a*** - numer matrycy 0 lub 1;

***x*** - znaki szesnastkowe 0..F składające się na dotychczasowe hasło *WRITE0/1*;

***y*** - znaki szesnastkowe 0..F składające się na nowe hasło *WRITE0/1.*

Polecenie umożliwia zmianę dotychczasowego hasła zabezpieczającego matryce *Array0* i *Array1* przed zapisem.

## Zmiana haseł odczytu

*Składnia polecenia:*

Cr***a***:***xxxxxxxxxxxxxxxx***:***yyyyyyyyyyyyyyyy***

***a*** - numer matrycy 0 lub 1;

***x*** - znaki szesnastkowe 0..F składające się na dotychczasowe hasło *READ0/1*;

***y*** - znaki szesnastkowe 0..F składające się na nowe hasło *READ0/1*.

Polecenie umożliwia zmianę dotychczasowego hasła zabezpieczającego matryce *Array0* i *Array1* przed odczytem.

## Zmiana hasła zabezpieczającego dostęp do poleceń *RESET*

*Składnia polecenia:*

Crr:***xxxxxxxxxxxxxxxx***:***yyyyyyyyyyyyyyyy***

***x*** - znaki szesnastkowe 0..F składające się na dotychczasowe hasło *RESET*;

***y*** - znaki szesnastkowe 0..F

składające się na nowe hasło *RESET*.

Polecenie umożliwia modyfikację hasła zabezpieczającego dostęp do poleceń zerowania rejestrów haseł, licznika błędnych prób i matryc pamięciowych.

## Zerowanie matryc pamięciowych oraz rejestrów haseł

*Składnia polecenia:*

DRS:**xxxxxxxxxxxxxxxx**

***x*** - znaki szesnastkowe 0..F składające się na hasło *RESET*.

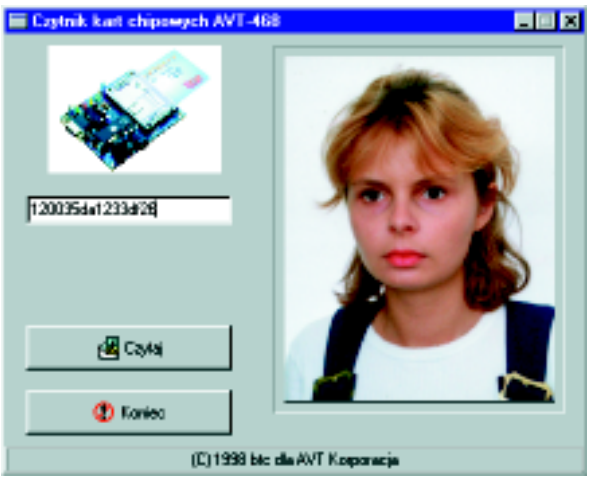
Polecenie umożliwia wyzerowanie licznika błędnych prób, obydwu matryc pamięciowych oraz rejestrów haseł.

Poprawne wykonanie polecenia sygnalizowane jest przez programator tekstowym potwierdzeniem „OK...“. W przypadku błędu składni polecenia wysyłany jest komunikat o przypuszczalnym błędzie.

Przy pomocy zestawu poleceń przedstawionego powyżej możliwe jest wykonanie praktycznie wszystkich operacji na karcie z poziomu programu terminalowego. Implementacja tych poleceń w dowolnym programie narzędziowym, napisanym specjalnie z myślą o współpracy z czytnikiem umożliwia osiągnięcie bardzo interesujących efektów - przykładem niech będzie program obsługujący sterowany elektronicznie zamek z identyfikacją osób wchodzących.

## Przykładowa aplikacja

Dzięki uniwersalnej konstrukcji karty chipowe mogą znaleźć cały szereg zastosowań. Jednym z najbardziej oczywistych jest elektroniczna kontrola dostępu. W laboratorium AVT powstało proste oprogramowanie dla komputera PC (Windows 95) sterujące pracą czytnika (**rys. 12**), które można wykorzystać do rejestracji i identyfikacji osób wchodzących do chronionego pomieszczenia.




Rys. 12. Okno programu wyświetlającego zdjęcie użytkownika karty.

Mniejszą matrycę pamięciową *Array0* wykorzystano do przechowywania numeru identyfikującego posiadacza karty. Matrycę pamięciową *Array1* wykorzystano do przechowywania zdjęcia użytkownika karty. Plik zawierający zdjęcie musi mieć objętość 8192B i powinien być zapisany w standardzie JPG. Przy pomocy osobnego okna prezentowanego programu możliwy jest zapis identyfikatora oraz zdjęcia do karty chipowej.

Prezentowane oprogramowanie będzie wchodziło w skład zestawu AVT-468.

**Piotr Zbysiński, AVT**

*Autor zastrzega możliwość modyfikacji programu sterującego pracą czytnika - programatora. Informacje o wprowadzonych modyfikacjach będą dołączane do zestawów AVT-468.*

*Trwają prace nad prostym, autonomicznym czytnikiem kart chipowych, które będzie można stosować jako stacjonarne sterowniki zamków elektromagnetycznych. Projekt takiego urządzenia przedstawimy w jednym z najbliższych numerów EP.*