Elektroniczny klucz do PC, część 3

kit AVT-330

Opis oprogramowania karty elektronicznego klucza do PC można było zakończyć na przedstawionym w drugiej części artykułu programie asemblerowym. Pozostałby jednak pewien niedosyt, dlaczego na przykład zmiana kodu kluczy musi wymagać ingerencji we wnętrze komputera, aby zewrzeć ze sobą odpowiednie punkty na trudno dostępnej płytce drukowanej. W trzeciej części artykułu opisano dołączane do karty oprogramowanie pracujące pod kontrolą systemu MS-DOS, usuwające między innymi wspomnianą niedogodność. Omówione będą również dodatkowe programy (dołączane do kitu) wspomagające przygotowanie pliku binarnego BIOS-u.

Informacje przydatne dla programistów

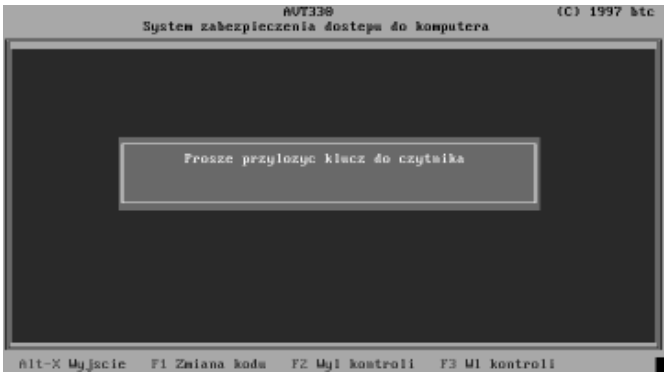
Przygotowanie pliku binarnego dla rozszerzenia BIOS-u wydawałoby się sprawą prostą - wystarczy wywołać odpowiedni kompilator i gotowe. Niestety, nie jest to takie proste. Mikroprocesory rodziny 80x86 niezbyt często występują w amatorskich konstrukcjach, toteż zdobycie asemblera generującego plik wyjściowy w formacie akceptowalnym przez programatory pamięci EPROM może być trudne (lub raczej kosztowne). Najczęściej spotykanymi kompilatorami asemblera dla procesorów 80x86 są TASM i MASM. Są to jednak kompilatory przeznaczone do tworzenia programów pracujących pod kontrolą systemu MS-DOS. Stosowanie tych kompilatorów ma istotną zaletę - można wykorzystać potężne narzędzia wspomagające proces uruchamiania pisanego programu.

Trudności związane z utworzeniem pliku binarnego można w sposób dosyć prosty obejść. Wystarczy wygenerować plik wynikowy typu COM, który w odróżnieniu od plików typu EXE, nie zawiera nagłówka i nie wprowadza podziału pamięci na segment danych i segment kodu. Plik taki mógłby być naszym plikiem binarnym, z jednym zastrzeżeniem - programy typu COM w systemie MS-DOS są wykonywane od adresu startowego 100h. Tak więc pierwszy bajt pliku zawiera kod instrukcji, który przy uruchomieniu takiego programu jest ładowany pod adres 100h, przy czym 100h nie jest adresem bezwzględnym, lecz ofsetem w 64kB segmencie pamięci. Jest to sytuacja dla nas dogodna, bo plik COM można dołączyć do 256-bajtowego pliku zawierającego dane wymagane w specyfikacji rozszerzenia BIOS-u. Pliki te można połączyć np. rozkazem:

```
copy /b pocz4.bin+program.com out.bin.
```

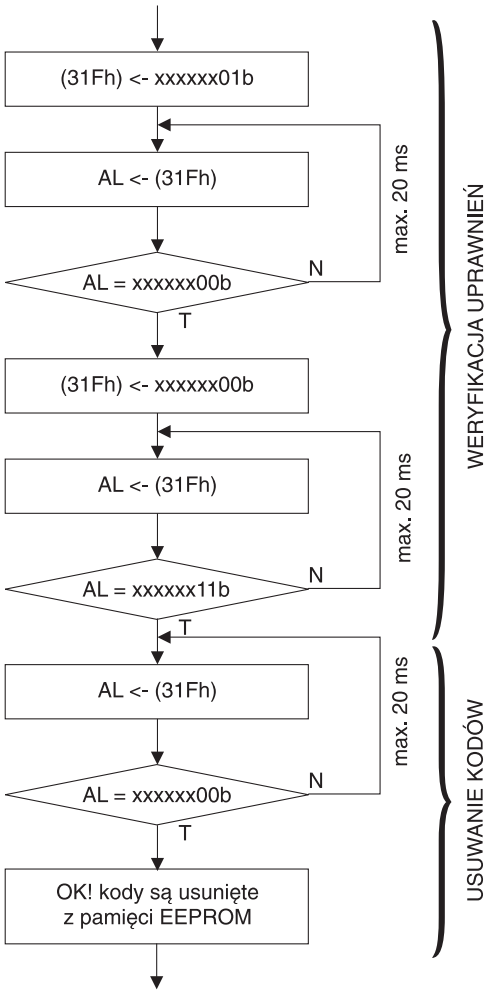
nr bajtu	wartość	znaczenie
0	55h	nagłówek
1	AAh	nagłówek
2	08h	liczba 512-bajtowych bloków rozszerzenia BIOS-u
3	E9h	rozkaz JMP 100h
4	FAh	
5	FFh	
.....		
255	FFh	ostatni bajt nagłówka

Rys. 6. Struktura przykładowego pliku nagłówkowego.



Rys. 7. Menu programu AVT330.EXE.

Struktura przykładowego pliku nagłówkowego, przy założonej wielkości rozszerzenia BIOS-u równej 4kB, jest pokazana na **rys.6**. Na dołączanej do kitu dyskietce znajdują się pliki nagłówkowe o nazwach nagl2.bin, nagl4.bin, nagl6.bin, nagl8.bin, które mogą być wykorzystane do tworzenia własnych wersji rozszerzenia BIOS-u (liczba w nazwie określa długość programu, a nie wielkość pamięci EPROM, do której będzie zapisany program).



Rys. 8. Algorytm procedury zmiany kodu.

Na tym jednak nie koniec. Należy spełnić jeszcze jeden warunek: suma modulo 100h wartości wszystkich bajtów rozszerzenia BIOS-u musi być równa 0. Na **listingu 3** przedstawiono program w języku C, który rozwiązuje ten problem. Parametrami wejściowymi dla programu są:

- nazwa wejściowego pliku binarnego;
- wielkość programu rozszerzenia BIOS w kilobajtach (dopuszczalne wartości: 2, 4, 6, 8);
- wielkość pamięci EPROM w kilobajtach, dla której tworzony jest plik wynikowy (dopuszczalne wartości: 2, 4, 8).

Po kontroli parametrów wywołania, wejściowy plik binarny jest wpisywany do tablicy o nazwie tablica, a następnie tablica jest uzupełniana wartościami FFh, aż do ostatniego bajtu programu, gdzie wpisywana jest wartość korygująca tak, żeby suma modulo 100h była równa 0. Tak utworzona tablica jest zapisywana do pliku pod nazwą out.bin. Skompilowany do postaci wykonywalnej program z listingu 3 znajduje się na dyskietce dołączanej do kitu (program BMAKE.EXE). Dodatkowo na dyskietce znajduje się program BCHECK.EXE, przeprowadzający kontrolę pliku, który ma zawierać kod rozszerzenia BIOS-u. Kontrolowany jest bajt określający długość programu oraz obliczana jest suma modulo 100h wartości zapisanych w pliku. Wywołanie programu wygląda następująco:

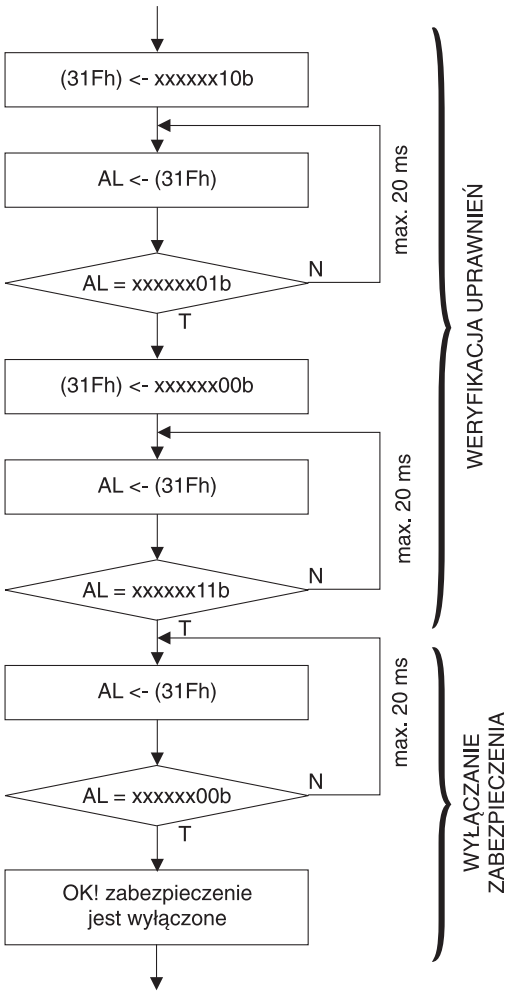
bcheck plik.bin

gdzie plik.bin jest nazwą pliku wejściowego.

Podsumowując, aby otrzymać plik binarny zawierający program rozszerzenia BIOS-u należy postępować zgodnie ze schematem zawartym w **tabeli 1** (przy wykorzystaniu TASM i TLINK firmy Borland).

Listing 3. Program wspomagający tworzenie pliku binarnego dla programatora.

```
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
  FILE *plik, *plik_wy;
  int kod, eprom; // 2|4|6|8
  int c, modulo, i;
  unsigned long suma, licznik_bajtow;
  int tablica[8200];

  printf( "\nPrzygotowanie BIOS-u\n" );
  if (argc != 4)
  {
    fprintf(stderr,"\nWywolanie:\n    \
bmake.exe plik_we [2|4|8] [2|4|8]\n");
    fprintf(stderr,"                    \
^kod    ^eprom\n");
    return 1;
  } // if argc != 2...

  if ( (plik = fopen(argv[1],"rb")) == NULL )
  {
    printf( "Blad otwarcia pliku z|BIOS-em"
);
    sound( 100 );  delay( 300 );  nosound();
    return 0;
  }

  if ( (plik_wy = fopen("out.bin", "wb"))  \
== NULL )
  {
    printf( "Blad otwarcia pliku \
wyjsciowego" );
    sound( 100 );  delay( 300 );  nosound();
    fclose( plik );
    return 0;
  }

  kod = atoi( argv[2] );
  switch( kod )
      {
        case 2|: { kod = 2048; break; }
        case 4|: { kod = 4096; break; }
        case 6|: { kod = 6144; break; }
        case 8|: { kod = 8192; break; }

        default : {
                    printf( "\n\n\n\n\n   \
Blednie podana wielkosc kodu!\n" );
                    fclose( plik );
                    fclose( plik_wy );
                    return 1;
                  }
      }

  eprom = atoi( argv[3] );
  switch( eprom )
      {
        case 2|: { eprom = 2048; break; }
        case 4|: { eprom = 4096; break; }
        case 8|: { eprom = 8192; break; }

        default : {
                    printf( "\n\n\n\n\n  \
Blednie podana wielkosc EPROM!\n" );
                    fclose( plik );
                    fclose( plik_wy );
                    return 1;
                  }
      }

  // inicjacja tablicy
  for( i=0; i<8199; i++)
    tablica[i] = 255;

  // przepisanie programu do tablicy
  licznik_bajtow = 0;
  while( (c = fgetc(plik)) != EOF )
  {
    tablica[licznik_bajtow] = c;
    licznik_bajtow++;
    if( licznik_bajtow > kod )
    {
      printf( "\nPrzesadziles troche   \
z|dlugoscia tego pliku\n" );
      fclose( plik ); fclose( plik_wy );
      return 0;
    }
  } // while

  // obliczanie sumy mod 100h
  suma = 0;
  licznik_bajtow = 0;
  modulo = 0;

  for( i=0; i<kod-1; i++)
  {
    suma += tablica[i];
    modulo += tablica[i];
    if( modulo > 255 )
      modulo -= 256;
  }

  tablica[kod-1] = 256 - modulo;

  for( i=0; i<eprom; i++)
    fputc( tablica[i], plik_wy );

  fclose( plik );
  fclose( plik_wy );
  return 0;
} // main()
```



Rys. 9. Algorytm procedury wyłączenia zabezpieczenia.

Zarządzanie trybem pracy karty z poziomu MS-DOS

Z dotychczasowego opisu możliwości funkcjonalnych karty elektronicznego klucza do PC można wnioskować, że przy każdym włączeniu komputera następuje prośba o przyłożenie pastylki DS1990 do czytnika. Na szczęście tak jednak nie jest! Można przecież wyobrazić sobie sytuację, w której musimy umożliwić komuś pracę na naszym komputerze, a nie chcemy wręczać danej osobie pastylki DS1990 (może ona być niezbędna do uruchomienia naszego samochodu lub włączenia/wyłączenia systemu alarmowego w naszym domu). W takiej sytuacji nie ma potrzeby wyjmowania karty z komputera, wystarczy uruchomić odpowiedni program, który po weryfikacji uprawnień wyłączy zabezpieczenie. Na dyskietce dołączanej do kitu jest umieszczony program (plik AVT330.EXE), którego menu jest pokazane na **rys. 7**. W programie zaimplementowano trzy procedury podnoszące walory użytkowe karty.

Tabela 1.

Krok	Operacja	Wynik
1	tasm program.asm	program.obj
2	tlink /t program.obj	program.com
3	copy /b nagl4.bin+program.com temp.bin	temp.bin
4	bmake temp 4 8	out.bin

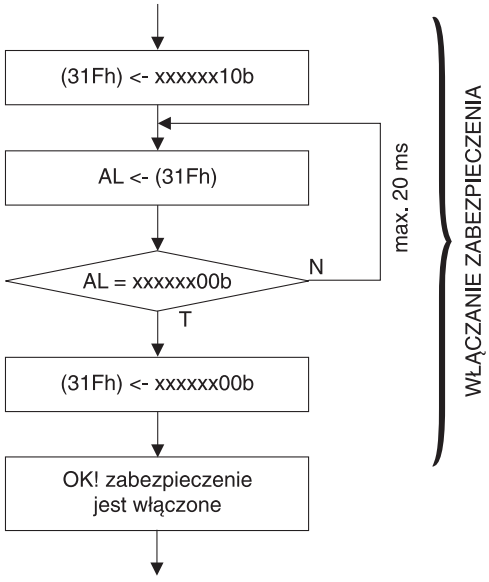
✗ **Zmiana kodu**
Wywołanie tej procedury następuje po wciśnięciu klawisza F1. Nazwa procedury jest trochę myląca, gdyż nie powoduje ona zmiany kodu, lecz wymazanie kodów z pamięci EEPROM. Bezpośrednią tego konsekwencją jest konieczność wprowadzenia nowych kodów przy następnym uruchomieniu komputera. Na **rys. 8** jest pokazany algorytm realizujący usuwanie kodów z pamięci EEPROM. W pierwszej fazie do mikrokontrolera US3 (port 31Fh) jest wysyłany bajt o wartości xxxxxx01b (x oznacza dowolną wartość). Wartość ta informuje mikrokontroler o zainicjowaniu procedury usuwania kodów, mikrokontroler potwierdza odebranie tej informacji zwracając bajt xxxxxx00b. W tym momencie na ekranie komputera jest wypisywany komunikat: „Proszę przyłożyć klucz do czytnika“ i mikrokontroler przechodzi w tryb oczekiwania na przyłożenie pastylki DS1990 do czytnika. Odczytanie prawidłowego kodu jest sygnalizowane wysłaniem przez mikrokontroler bajtu o wartości xxxxxx11b, po czym następuje wymazanie kodów z pamięci. Jeżeli procedura usuwania kodów zakończy się pomyślnie, mikrokontroler informuje o tym wysyłając bajt o wartości xxxxxx00b.

✗ **Wyłączenie kontroli**
Algorytm wyłączenia kontroli (**rys. 9**) jest bardzo podobny do algorytmu zmiany kodu. Również w tym przypadku, w pierwszej kolejności jest przeprowadzana weryfikacja uprawnień. Procedury te różnią się jedynie wartościami wymienianymi między komputerem PC a kartą. Od momentu wyłączenia kontroli nie będzie konieczne przykładanie pastylki DS1990 przy uruchamianiu komputera. Jedynym przejawem zainstalowania karty będzie komunikat wyświetlany przy procedurze POST. Procedura jest uruchamiana po przyciśnięciu klawisza F2.

✗ **Włączenie kontroli**
Algorytm tej procedury pokazano na **rys. 10**. Procedura ta jest wywoływana po przyciśnięciu klawisza F3. Przy włączaniu kontroli, z oczywistych powodów, nie jest wymagane posiadanie klucza. W pierwszym kroku do mikrokontrolera jest wysyłany bajt o wartości xxxxxx10b. Jest to taka sama wartość jak przy wyłączaniu kontroli, więc następne działania są podejmowane w zależności od stanu, w jakim znajduje się mikrokontroler US3. Włączenie kontroli jest potwierdzane przez mikrokontroler wysłaniem wartości xxxxxx00b.

Działanie omówionych wyżej procedur można przerwać wciskając klawisz ESC. Zakończenie pracy programu następuje po wciśnięciu Alt-X.

Paweł Zbysiński



Rys. 10. Algorytm procedury włączenia zabezpieczenia.