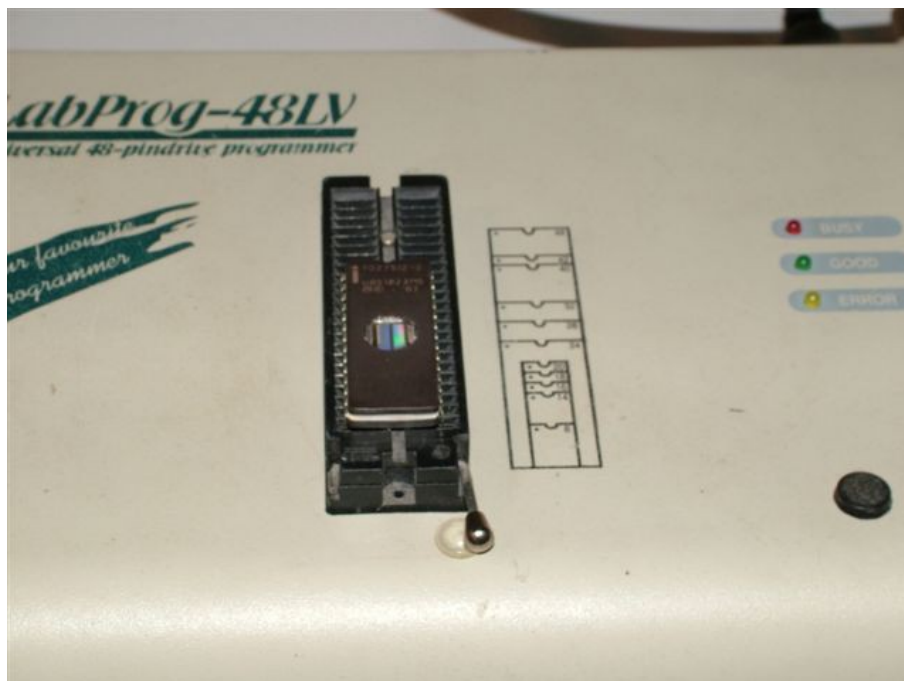


Andrzej Pawluczuk



Filozofia sieci: Format pliku Intel-hex

*dla Elportal
Białystok, styczeń 2019*

Dane będące wynikowym programem dla mikroprocesorów i mikrokontrolerów mogą być zapisywane w pliku na wiele sposobów. Znanych jest kilka formatów, z których najpopularniejszy jest format szesnastkowy opracowany w firmie Intel (stąd jego nazwa Intel-hex). W pliku tym występują znaki alfanumeryczne oraz znaki *CR* oraz *LF* (jako znaki sterujące nie mające swej reprezentacji graficznej).

W przypadku programów, które nie przekraczają 64kB wielkości kodu, plik Intel-hex składa się w dwóch rodzajów rekordów:

- rekordu danych,
- rekordu końca pliku.

W obu przypadkach każdy rekord rozpoczyna się od znaku ":", po którym następują odpowiednie dane (wszystkie dane są zapisywane w notacji szesnastkowej, tj. po dwa znaki alfanumeryczne do zakodowania binarnej wartości 8-bitowej):

- dwa znaki określające długość rekordu,
- cztery znaki określające adres w pamięci pod którym należy umieścić dane,
- dwa znaki określające typ danych,
- dane (po dwa znaki na każdy bajt zapisywany w pamięci),
- dwa znaki sumy kontrolnej.

Każdy rekord zakończony jest dwoma znakami sterującymi (*CR* o kodzie 0D hex i *LF* 0A hex) określającymi koniec wiersza.

Występujący w zbiorze typ danych może być:

- 00 hex (0 binarnie) – jako oznaczenie rekordu zawierającego dane do zapisu w pamięci,
- 01 hex (1 binarnie) – jako oznaczenie rekordu kończącego plik.

Suma kontrolna (czasami błędnie nazywana sumą CRC) jest 8-bitową sumą wszystkich danych zawartych w rekordzie pliku.

Jedną z istotnych cech tego formatu jest opis zawartości tylko istotnych danych. W ogólnym przypadku nie ma obowiązku, by zapisany w pliku blok danych stanowił lity, ciągły obszar. Możliwe jest, by plik zawierał „dziury”. Oczywiście jest to oddzielna kwestia związana z interpretacją takich „dziur”. Sensownym wyjaśnieniem ich istnienia jest początkowy obszar kodu z programem dla mikrokontrolera z rodziny C51. Wiadomo, że ten mikrokontroler w pewnym obszarze zawiera procedury obsługi przerwań, które są umieszczane w ściśle określonym miejscu przestrzeni adresowej. Na przerwanie przewidziane jest kilka bajtów. Nie ulega wątpliwości, że w tak małym obszarze nie jest możliwe zaimplementowanie procedury obsługi przerwania, toteż umieszcza się w dedykowanym obszarze do obsługi przerwania instrukcję skoku „w głąb” przestrzeni adresowej. Niewykorzystane komórki (takie, w których kompilator w parze z linkerem nie umieścił w nich żadnych kodów) stanowią „dziury”. Poniższy przykład opisuje blok pamięci z „dziurami” (choć często wygenerowany kod stanowi ciągły pod względem adresowym obszar danych).

Przykładowy kod hex (we fragmentach):

```
:03000000C38000BA
:03000800C36900C9
...
:10006600C37E00FBED4DFBED4DFBED4DFBED4DFB7A
...
:1000E60073756D6965206D612062796320316B429D
:00000001FF
```

Pierwszy rekord zawiera:

:03000000C38000BA

: - początek rekordu,

03 hex = 3 dziesiętnie – 3 bajty danych do pamięci,

0000 hex = 0 dziesiętnie – miejsce w pamięci do zapisu danych (patrz ORG 0),

00 hex = 0 dziesiętnie – typ danych ==> dane do zapisu w pamięci,

C3 hex, 80 hex, 00 hex – 3 kolejne bajty danych do zapisu w pamięci,

BA hex = 186 dziesiętnie – suma kontrolna obliczona jako:

03	hex	=	003	dec
00	hex	=	000	dec
00	hex	=	000	dec
00	hex	=	000	dec
C3	hex	=	195	dec
80	hex	=	128	dec
00	hex	=	000	dec
+	BA	hex	=	186

200		hex	=	512
			dec	

Po obcięciu do 8 bitów 200 hex ==> 00 hex (512 mod 256 również daje 0) suma musi wynosić 0.

Do kontroli wystarczy zsumować odpowiednie liczby i sprawdzić, czy wynik wynosi zero. Do wygenerowania sumy należy obliczyć ją następująco:

03	hex	=	003	dec	
00	hex	=	000	dec	
00	hex	=	000	dec	
00	hex	=	000	dec	
C3	hex	=	195	dec	
80	hex	=	128	dec	
+	00	hex	=	000	dec

146		hex	=	326	dec

146 hex po obcięciu daje 46 hex = 70 dec (lub 326 / 256 daje 1 i 70 reszty).

Od liczby 100 hex (256 dec) należy odjąć obliczoną sumę:

100 hex – 46 hex = BA hex (186 dec)

lub 256 dec – 70 dec = 186 dec.

Drugi rekord zawiera:

:03000800C36900C9

: - początek rekordu,

03 hex = 3 dziesiętnie – 3 bajty danych do pamięci,

0008 hex = 8 dziesiętnie – miejsce w pamięci do zapisu danych (patrz ORG 8H),

00 hex = 0 dziesiętnie – typ danych ==> dane do zapisu w pamięci,

C3 hex, 69 hex, 00 hex – 3 kolejne bajty danych do zapisu w pamięci,

C9 hex = 201 dziesiętnie – suma kontrolna obliczona jako:

03	hex	=	003	dec	
00	hex	=	000	dec	
08	hex	=	008	dec	
00	hex	=	000	dec	
C3	hex	=	195	dec	
69	hex	=	105	dec	
00	hex	=	000	dec	
+	C9	hex	=	201	dec

200		hex	=	512	dec

Po obcięciu do 8 bitów 200 hex ==> 00 hex = 0 dec.

Inny rekord zawiera:

:10006600C37E00FBED4DFBED4DFBED4DFBED4DFB7A

: - początek rekordu,

10 hex = 16 dziesiętnie – 16 bajtów danych do pamięci,

0066 hex = 102 dziesiętnie – miejsce w pamięci do zapisu danych,

00 hex = 0 dziesiętnie – typ danych ==> dane do zapisu w pamięci,

C3 hex, 7E hex, 00 hex, FB hex, ED hex, 4D hex, FB hex, ED hex, 4D hex, FB hex, ED hex, 4D hex, FB hex, ED hex, 4D hex, FB hex – 16 kolejnych bajtów danych do zapisu w pamięci,

7A hex = 122 dziesiętnie – suma kontrolna obliczona jako:

10	hex	=	016	dec
00	hex	=	000	dec
66	hex	=	102	dec
00	hex	=	000	dec
C3	hex	=	195	dec
7E	hex	=	126	dec
00	hex	=	000	dec
FB	hex	=	251	dec
ED	hex	=	237	dec
4D	hex	=	077	dec
FB	hex	=	251	dec
ED	hex	=	237	dec
4D	hex	=	077	dec
FB	hex	=	251	dec
ED	hex	=	237	dec
4D	hex	=	077	dec
FB	hex	=	251	dec
ED	hex	=	237	dec
4D	hex	=	077	dec
FB	hex	=	251	dec
ED	hex	=	237	dec
4D	hex	=	077	dec
FB	hex	=	251	dec
7A	hex	=	122	dec

C00	hex	=	3072	dec

Po obcięciu do 8 bitów C00 hex ==> 00 hex = 0 dec.

Ostatni rekord:

```
:00000001FF
```

: - początek rekordu,

00 hex = 0 dziesiętnie – rekord nie zawiera danych do pamięci,

0000 hex = 0 dziesiętnie – miejsce w pamięci do zapisu danych (pole w sumie nieistotne),

01 hex = 01 dziesiętnie – typ danych ==> zakończenie pliku,

<brak danych>

FF hex = 255 dziesiętnie – suma kontrolna obliczona jako:

	00	hex	=	000	dec
	00	hex	=	000	dec
	00	hex	=	000	dec
	01	hex	=	001	dec
+	FF	hex	=	255	dec

	100	hex	=	256	dec

Po obcięciu do 8 bitów 100 hex ==> 00 hex = 0 dec.

Ale niestety, życie przerosło kabaret, tak jak pojemności pamięci przekroczyły barierę 64kB. Przykładowa wczytana zawartość pamięci BIOS komputera PC z procesorem PENTIUM czterokrotnie przekracza możliwości 16-bitowego adresowania. Coś z tym trzeba było zrobić, więc standard został rozszerzony o nowy rodzaj rekordu, rzeczywistość nie znosi luk i zawsze zawsze da się coś wymyślić. Dodany został typ rekordu o wartości kodowej 4, który informuje o numerze bloku o adresacji 16-bitowej (stanowi rozszerzenie adresu).

Zawartość pamięci BIOS we fragmentach:

```
:020000040000FA
:1000000025202D6C68352D463E0000405A0000002A
:1000100000004120010C61776172646578742E7272
:100020006F6D54E12000002CA88F77BF6B1252EE49
...
:10FFC000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF41
:10FFD000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF31
:10FFE000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF21
:10FFF000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF11
:020000040001F9
:10000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00
:10001000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00
:10002000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
...
:10FFE000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF21
:10FFF000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF11
:020000040002F8
:1000000025052D6C68352D2C5D0100000020000D7
:1000100000005020010C41423631533031352E4220
:10002000494ECDD22000002D908F01F6B1252ED162
...
:10FFD0005CC687413AD1A67B4199CD60A5A2CBED05
```

```

:10FFE0000F4F6A68C327883D3D49A5D5EAB4DF7E37
:10FFF00029F859DF04859E7F413B9E9AF084D3C245
:020000040003F7
:10000000CBCD34682F0A6A77379A6826935A2D0821
:1000100031CFDE2BF859FD5EA95C56FBF0A5A5F1AA
....
:10FFE000000000000000000000000000324136394B48324921
:10FFF000EA5BE000F02A4D52422A02000000FFFFB7
:020000040004F6
:00000001FF

```

Rekord rozszerzenia adresu:

```
:020000040000FA
```

: - początek rekordu,

02 hex = 2 dziesiętnie – wielkość w bajtach informacji

0000 hex = 0 dziesiętnie – miejsce w pamięci do zapisu danych (pole w sumie nieistotne),

04 hex = 4 dziesiętnie – typ danych numer bloku o wielkości 64kB

0000 hex = 0 dziesiętnie – numer 64kB-owego bloku

FA hex = 250 dziesiętnie – suma kontrolna obliczona jako:

02	hex	=	002	dec
00	hex	=	000	dec
00	hex	=	000	dec
04	hex	=	004	dec
00	hex	=	000	dec
00	hex	=	000	dec
+	FA	hex	=	250

100	hex	=	256	dec

Po obcięciu do 8 bitów 100 hex ==> 00 hex = 0 dec.

No i zniknęły dotychczasowe ograniczenia, można „wypłynąć na szerokie wody”.