

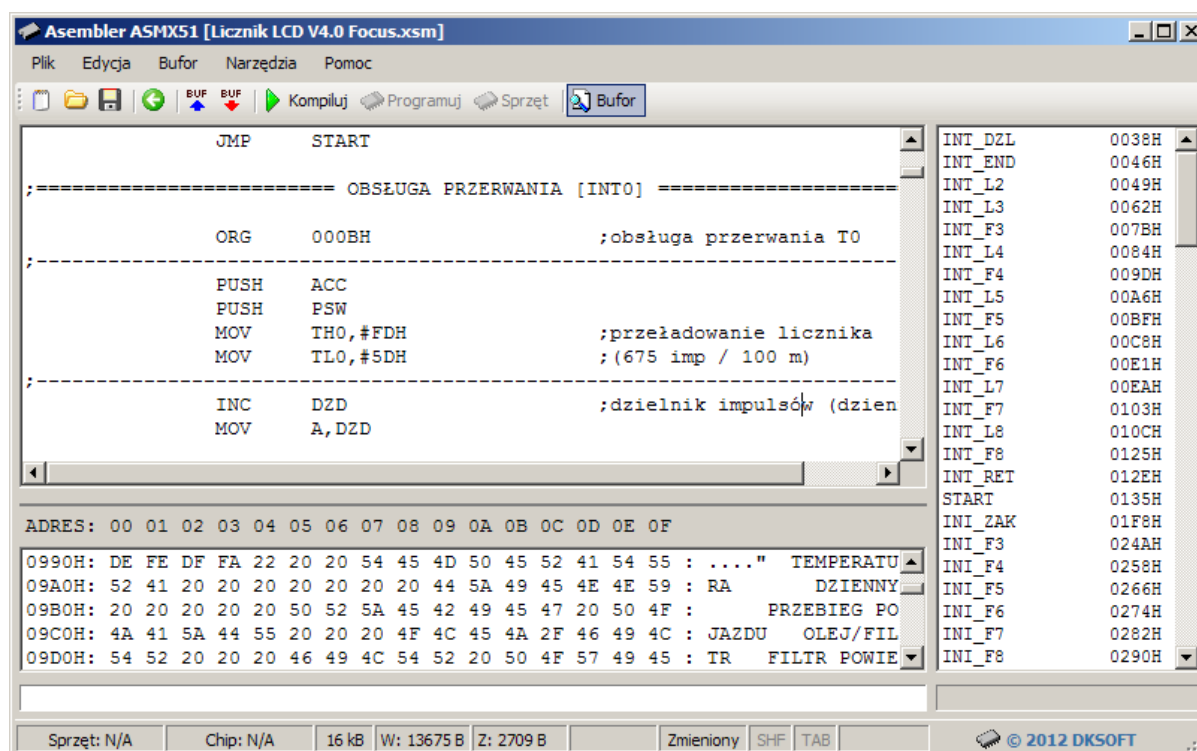
Dariusz Kozak

ZESTAW URUCHOMIENIOWY  
MIKROKOMPUTERÓW JEDNOUKŁADOWYCH MCS-51

ZUX51

# Asembler ASMX51

INSTRUKCJA OBSŁUGI



© 2012 DK Wszystkie prawa zastrzeżone

Kopiowanie, powielanie i rozpowszechnianie w jakiegokolwiek formie bez zgody autora jest zabronione. Wszystkie nazwy stanowiące znaki firmowe i/lub towarowe użyte w tekście są własnością odpowiednich podmiotów.

## 1. WPROWADZENIE

Asembler ASMX51 jest prostym i łatwym w użyciu w pełni funkcjonalnym narzędziem programistycznym dla procesora 8051. Posiada zaimplementowaną pełną listę rozkazów (z kilkoma wyjątkami opisanymi niżej), ograniczoną do niezbędnych liczbę dyrektyw oraz ustalone sposoby zapisu argumentów.

### ZMIANY:

(w stosunku do oryginalnej listy rozkazów oraz standardowych assemblerów MCS-51)

- usunięto instrukcje skoków i wywołań stronicowanych AJMP i ACALL
- zmieniono zapis instrukcji ANL i ORL z zanegowanym argumentem na NANL i NORL (przykład):
  - `NANL C,20H` zamiast `ANL C,/20H`
  - `NORL C,20H` zamiast `ORL C,/20H;`
- zmieniono zapis argumentu adresowanego pośrednio (przykład):
  - `MOV A,[A+DPTR]` zamiast `MOV A,@A+DPTR`
  - `MOV [R1],#FFH` zamiast `MOV @R1,#FFH`
  - `JMP [A+DPTR]` zamiast `MOV @A+DPTR`
- Nie są obsługiwane operacje arytmetyczne ani logiczne bezpośrednio na tekście źródłowym

### ARGUMENTY:

Dopuszczalne kombinacje formatów zapisu z argumentami instrukcji:

Operand:	Dopuszczalny argument:
ADR16 (adres 16-bitowy)	Etykieta ADR
#NN (stała 16-bitowa)	Etykieta ADR, HEX16, DEC16
ADR8 (adres 8-bitowy)	HEX08, BIN08, DEC08, Etykieta EQU
#N (stała 8-bitowa)	HEX08, BIN08, DEC08, Etykieta EQU
REL (przesunięcie)	Etykieta ADR

gdzie:

Etykieta ADR	etykieta adresowa (wartość 16-bitowa)
Etykieta EQU	etykieta stałej (wartość 8-bitowa)
HEX16	liczba 16-bitowa, format HEX (np. 01FFH)
DEC16	liczba 16-bitowa, format DEC (np. 16787)
HEX08	liczba 8-bitowa, format HEX (np. FFH)
BIN08	liczba 8-bitowa, format BIN (np. 01101111B)
DEC08	liczba 8-bitowa, format DEC (np. 255)

### DYREKTYWY:

Dyrektywy ograniczono się do następujących:

ORG	organizacji kodu
EQU	deklaracji stałych
DB, DW, DS	włączania danych

oraz (tylko procesor AT89S8253):

ORGEEP, ORGUSG	dyrektywa segmentu EEPROM i USERSG
----------------	------------------------------------

ORG - ustala początkowy adres w pamięci programu, od którego zostanie umieszczony kod występujący bezpośrednio po tej dyrektywie. Format argumentu: HEX16, DEC16 (przykład):

```
ORG 01FFH           ;argument HEX16
ORG 2047            ;argument DEC16
```

EQU - przypisuje etykietce 8-bitowy argument. Stosowana jako etykieta adresowa komórek pamięci RAM i rejestrów procesora: Format argumentu: HEX08, BIN08, DEC08, etykieta (przykład):

```
ETYKIETA EQU FFH           ;argument HEX8
ETYKIETA EQU 00001111B     ;argument BIN8
ETYKIETA EQU 255           ;argument DEC8
ETYKIETA EQU LICZNIK       ;etykieta EQU (wcześniej zadeklarowana)
ETYKIETA EQU ACC.7         ;etykieta PSS (stała etykieta systemowa)
```

DB - wprowadza argument 8-bitowy bezpośrednio w kod programu. Dopuszcza stosowanie wielu argumentów oddzielonych znakiem "przecinek". Format argumentu: HEX08, BIN08, DEC08, etykieta (przykład):

```
DB LICZNIK,ACC.7          ;etykieta EQU, etykieta PSS
DB FFH,00001111B,255     ;HEX08, BIN08, DEC08
```

DW - jak wyżej, lecz argumentem może być zarówno wartość 8-bitowa, jak i 16-bitowa. Format argumentu: HEX16, DEC16, HEX08, BIN08, DEC08, etykieta (przykład):

```
DW 1FFFH,LICZNIK,ACC.7    ;HEX16, etykieta EQU, etykieta PSS
DW 03FFH0,11110000B,23255 ;HEX16, BIN08, DEC16
```

(jeśli argumentem jest wartość 8-bitowa, zostanie zapisana na pozycji starszego bajtu. Młodszy bajt przyjmie wartość 00H)

DS - jak wyżej, lecz argumentem może być wyłącznie ciąg tekstowy, umieszczony pomiędzy znakami „apostrof” (przykład):

```
DS 'ciąg tekstowy'       ;kody ASCII kolejnych znaków (8-bitów/znak)
```

(pomiędzy apostrofami na początku i końcu ciągu, dopuszczalne są wszelkie znaki włącznie z pojedynczym lub wielokrotnym apostrofem)

ORGEEP - dyrektywa organizacji kodu dla pamięci EEPROM procesora AT89S8253. Dyrektywa obejmuje przestrzeń 2 kB pamięci od adresu 0000H do adresu 07FFH. Kod wynikowy umieszczany jest w buforze programu zaraz za przestrzenią pamięci flash procesora (0000H - 2FFFH), czyli od adresu 3000H (poza buforem programu). Kod ten można wykorzystać do zaprogramowania przestrzeni pamięci EEPROM procesora, ale nie zostanie on ujęty podczas generowania pliku HEX lub BIN w czasie zapisu bufora programu na dysk. Przykład:

```
ORGEEP 0000H           ;kod rozpoczyna się od adresu 0000H
                       ;pamięci EEPROM (w buforze można go
                       ;zobaczyć pod adresem 3000H)
```

ORGUSG - dyrektywa organizacja kodu dla sygnatury użytkownika procesora AT89S8253. Obejmuje przestrzeń 64 B pamięci od adresu 0000H do 003FH. Kod wynikowy umieszczany jest w buforze programu zaraz za przestrzenią dyrektywy ORGE (3000H - 37FFH), czyli od adresu 3800H (poza buforem programu). Kod ten można

wykorzystać do zaprogramowania przestrzeni sygnatury użytkownika USERSG procesora, ale nie zostanie on ujęty podczas generowania pliku HEX lub BIN w czasie zapisu bufora programu na dysk. Przykład:

```
ORGUSG 0000H                ;kod rozpoczyna się od adresu 0000H
                             ;pamięci EEPROM (w buforze można go
                             ;zobaczyć pod adresem 3800H)
```

### ETYKIETY:

Etykieta ADR (adresowa) – musi rozpoczynać się literą, może składać się z maksymalnie z 16 znaków: liter, cyfr lub znaków „\_” oraz „-” (podkreślenie / myślnik). Etykieta nie może dublować innej etykiety, nazwy instrukcji, rejestru procesora, itp. Deklaracja etykiety adresowej musi być zakończona znakiem „dwukropek” (przykład):

```
START: MOV A,#255           ;etykieta adresowa
```

Etykieta EQU (stałej) – musi rozpoczynać się literą, może składać się z maksymalnie z 16 znaków: liter, cyfr lub znaków „\_” oraz „-” (podkreślenie / myślnik). Etykieta nie może dublować innej etykiety, nazwy instrukcji, rejestru procesora, itp. Po deklaracji etykiety stałej musi wystąpić dyrektywa EQU oraz 8-bitowy argument (przykład):

```
WEKTOR EQU 1FH             ;etykieta stałej
```

Etykieta PSS (predefiniowanej stałej systemowej) – etykieta zaimplementowana w programie na stałe, odnosząca się do rejestru bądź komórki procesora. Nazw tych etykiet nie można zmieniać, do wykorzystania jak wszystkie etykiety stałych (EQU). Wszystkie etykiety PSS obsługiwane przez program wymienia tabela 1.

### KOMENTARZE:

Komentarz musi rozpoczynać się od znaku „;” (średnik). Wszystkie znaki znajdujące się w prawo począwszy od tego znaku są ignorowane przez kompilator oraz kontroler składni edytora.

Komentarze można umieszczać w linii samodzielnie lub włączać je do instrukcji. Znakiem poprzedzającym komentarz jest „średnik” (przykład):

```
;Przykładowy, samodzielny komentarz (dostępna cała linia)
CZEKAJ: JNB WSK_1,CZEKAJ      ;komentarz do instrukcji
```

### POZOSTAŁE:

W assemblerze obowiązuje zapis wierszowy tj. wszystkie składniki instrukcji (etykieta, mnemonik, operand) odnoszące się do siebie muszą znajdować się w jednym wierszu.

Kompilator dopuszcza puste wiersze.

Nie są wymagane żadne dyrektywy początku i końca programu. Kompilacji podlega cały plik tekstowy znajdujący się w oknie edytora.

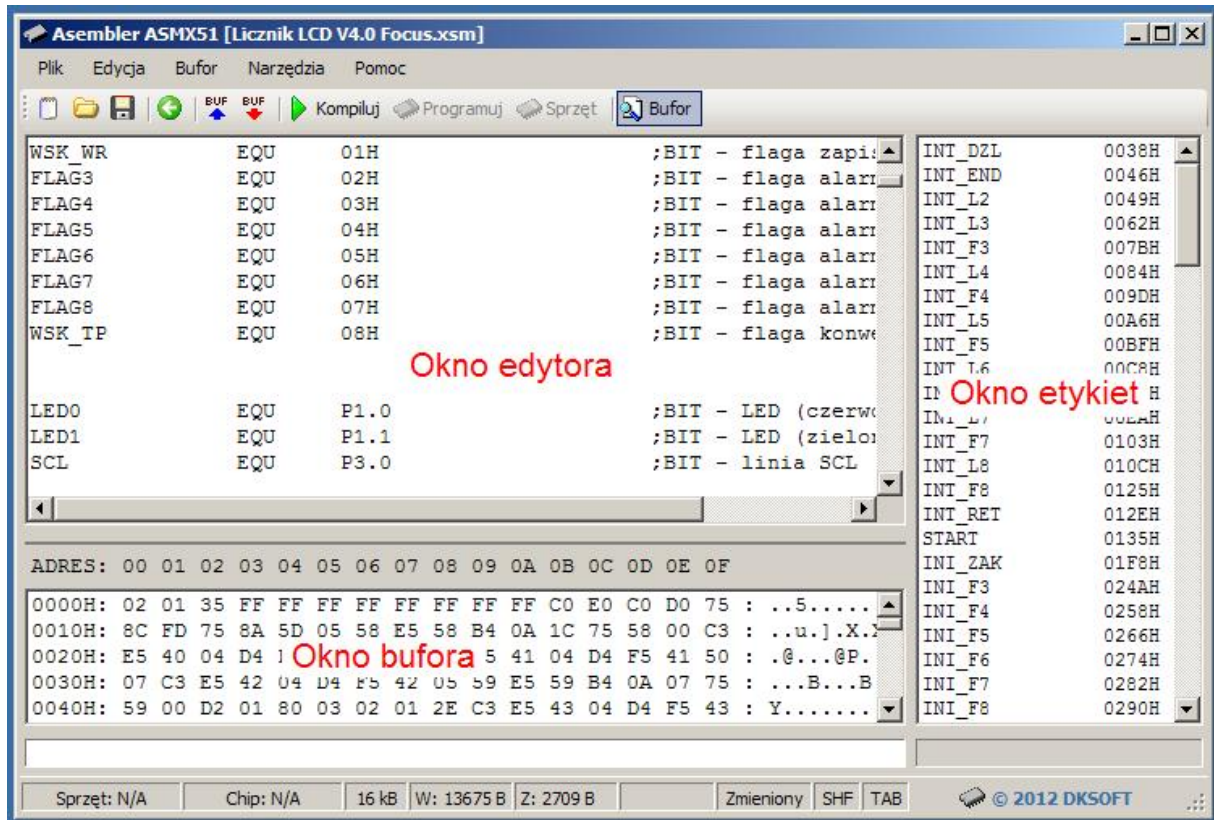
Kontroler składni sprawdza w czasie kompilacji poprawność wprowadzonego kodu i wskazuje ew. błędy zaznaczając wiersz pierwszego wystąpienia błędu wraz ze stosownym komentarzem.

PSS	ADR	Funkcja	+ 0	+ 1	+ 2	+ 3	+ 4	+ 5	+ 6	+ 7
ACC	E0H	akumulator	ACC.0	ACC.1	ACC.2	ACC.3	ACC.4	ACC.5	ACC.6	ACC.7
B	F0H	rejestr B	B.0	B.1	B.2	B.3	B.4	B.5	B.6	B.7
PSW	D0H	rejestr znaczników	P	-	OV	RS0	RS1	F0	AC	CY
SP	81H	wskaźnik stosu								
DPL	82H	wskaźnik danych DPTR - L								
DPH	83H	wskaźnik danych DPTR - H								
P0	80H	port 0	P0.0	P0.1	P0.2	P0.3	P0.4	P0.5	P0.6	P0.7
P1	90H	port 1	P1.0	P1.1	P1.2	P1.3	P1.4	P1.5	P1.6	P1.7
P2	A0H	port 2	P2.0	P2.1	P2.2	P2.3	P2.4	P2.5	P2.6	P2.7
P3	B0H	port 3	P3.0	P3.1	P3.2	P3.3	P3.4	P3.5	P3.6	P3.7
IE	A8H	rejestr priorytetów	EX0	ET0	EX1	ET1	ES	ET2*	-	EA
IP	B8H	rejestr przerw	PX0	PT0	PX1	PT1	PS	PT2*	-	-
TMOD	89H	rejestr trybu liczników								
TCON	88H	rejestr sterujący liczników	IT0	IE0	IT1	IE1	TR0	TF0	TR1	TF1
T2MOD*	C9H	rejestr trybu licznika T2								
T2CON*	C8H	rejestr sterujący licznika T2	CPRL2*	CT2*	TR2*	EXEN2*	TCLK*	PCLK*	EXF2*	TF2*
TL0	8AH	rejestr licznika T0 - L								
TH0	8CH	rejestr licznika T0 - H								
TL1	8BH	rejestr licznika T1 - L								
TH1	8DH	rejestr licznika T1 - H								
TL2*	CCH	rejestr licznika T2 - L								
TH2*	CDH	rejestr licznika T2 - H								
PCON	87H	rejestr ster. zasilaniem	SMOD	-	-	-	GF1	GF0	PD	IDL
SCON	98H	rejestr sterujący UART	RI	TI	RB8	TB8	REN	SM2	SM1	SM0
SBUF	99H	rejestr buforowy UART								
RCAP2L*	CAH	rejestr licznika T2 - L								
RCAP2H*	CBH	rejestr licznika T2 - H								
<b>Rejestry dodatkowe dla: 89S51/S52</b>										
DPL2	84H	wskaźnik danych DPTR2 - L								
DPH2	85H	wskaźnik danych DPTR2 - H								
AUXR	8EH	rejestr sterujący RST/WDT								
AUXR1	A2H	rejestr wyboru DPTR								
WDTRST	A6H	rejestr WATCHDOG								
<b>Rejestry dodatkowe dla: 89S8253</b>										
DPL2	84H	wskaźnik danych DPTR2 - L								
DPH2	85H	wskaźnik danych DPTR2 - H								
SPDR	86H	rejestr SPI								
AUXR	8EH	rejestr sterujący RST/WDT								
CLKREG	8FH	rejestr oscylatora x2								
EECON	96H	rejestr kontrolny EEPROM								
WDTRST	A6H	rejestr WATCHDOG								
WDTCON	A7H	rej. kontrolny WATCHDOG								
SADDR	A9H	rejestr adresu UART								
SADEN	B9H	rejestr maski UART								
IPH	B7H	rejestr przerw HIGH								
SPCR	D5H	rejestr sterujący SPI								
SPSR	AAH	rejestr statusu SPI								
* - rozszerzony licznik/timer T2 (89C52/S52/S8253)										

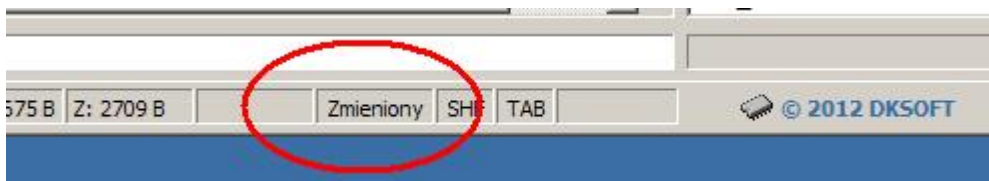
Tabela 1. Wykaz Rejestrów Specjalnego Przeznaczenia (SFR) oraz przypisanych no nich etykiet PSS zaimplementowanych w programie wraz z ich adresami w przestrzeni procesora

## 2. EDYTOR

Edytor składa się z trzech okien: właściwego okna edytora, okna etykiet adresowych oraz okna podglądu bufora. Poniżej okien znajdują się paski: komunikatów, postępu i statusu:



Zmiana zawartości okna edytora od czasu ostatniego zapisu, sygnalizowana jest na pasku statusu komunikatem „Zmieniony”:



Edytor wyposażony jest mechanizmy ułatwiające tworzenie przejrzystego kodu, poprzez możliwość włączenia automatycznej zmiany wielkości znaków (Auto SHF) i automatycznych tabulatorów (Auto TAB). Funkcje te opisane są niżej a ich aktywność wskazuje pasek statusu:





Funkcje dostępne dla edytora znajdujące się w menu oraz na pasku narzędziowym, opisano poniżej:

### **Nowy**

(Menu -> Plik -> Nowy / Pasek narzędziowy: tak)

Tworzy nowy, pusty plik programu.

### **Otwórz**

(Menu -> Plik -> Otwórz / Pasek narzędziowy: tak)

Otwiera plik programu z (rozszerzenie \*.xsm) i ładuje jego zawartość do okna edytora.

### **Zapisz**

(Menu -> Plik -> Zapisz / Pasek narzędziowy: tak)

Zapisuje zawartość okna edytora do pliku (rozszerzenie \*.xsm). Jeśli program nie posiada nazwy (nie był zapisany) otwiera okno dialogowe do zapisu. Dostępna jedynie, kiedy zawartość okna edytora uległa zmianie od czasu ostatniego zapisu.

### **Zapisz jako ...**

(Menu -> Plik -> Zapisz jako ... / Pasek narzędziowy: nie)

J/w, z tym, że zawsze otwiera okno dialogowe do zapisu. Umożliwia zapisanie zawartości okna edytora do pliku pod inną nazwą.

### **Cofnij**

(Menu -> Edycja -> Cofnij / Pasek narzędziowy: tak)

Dostępna jest tylko dla ostatniej operacji edycyjnej w oknie edytora

### **Szukaj/Zmień**

(Menu -> Edycja -> Szukaj/Zmień / Pasek narzędziowy: nie / Skrót: F3)

Pozwala wyszukać i zmienić wybrany ciąg znaków

### **Auto SHF**

(Menu -> Narzędzia -> Ustawienia / Pasek narzędziowy: nie)

Automatycznie włącza duże znaki po każdym złamaniu wiersza (klawiszem „ENTER”) i wyłącza je po wprowadzeniu znaku komentarza („;”).

### **Auto TAB**

(Menu -> Narzędzia -> Ustawienia / Pasek narzędziowy: nie)

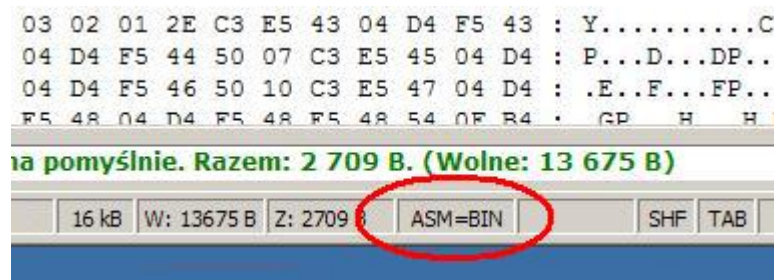
Automatycznie wprowadza określoną liczbę tabulatorów po złamaniu wiersza oraz po wprowadzeniu znaku komentarza („;”). Przy wprowadzaniu etykiety można użyć do łamania wiersza kombinacji klawiszy [SHIFT + ENTER] lub [CTRL + ENTER] aby pominąć jednorazowo wprowadzanie tabulatorów przed instrukcją. Liczba tabulatorów dla obydwu opcji ustala się w oknie „Ustawienia” (Menu: Narzędzia -> Ustawienia).

### **Parametry**

Pojemność okna edytora:	256 kB (znaków)
Pojemność bufora kodu:	16 kB
Max długość etykiety:	16 znaków

### 3. KOMPILATOR

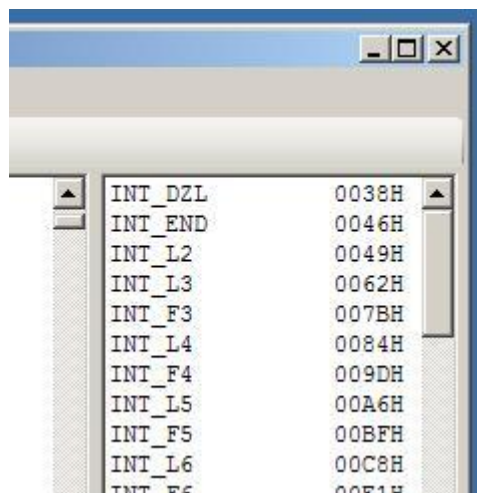
Program posiada kompilator dwuprzebiegowy. W pierwszym przebiegu tworzone są: tablice etykiet adresowych, stałych oraz przeprowadzana jest kontrola składni, w drugim następuje właściwa kompilacja programu źródłowego do kodu wynikowego. Kod będący wynikiem kompilacji umieszczany jest w buforze. Po zakończeniu kompilacji ustawiany jest znacznik zgodności oznaczający zgodność kodu wynikowego w buforze z plikiem źródłowym w oknie edytora, wprowadzenie jakiegokolwiek zmiany w oknie edytora kasuje ten znacznik:



Jeśli kompilator wykryje w czasie przetwarzania pliku źródłowego jakiegokolwiek błąd (składniowe, nieprawidłowe argumenty, przepełnienie bufora, itp.), następuje zatrzymanie procesu kompilacji, wyświetlenie błędu w pasku komunikatów oraz podświetlenie błędnego wiersza w oknie edytora.

#### Okno etykiet adresowych

Po prawidłowo przeprowadzonej kompilacji, wypełniane jest okno etykiet adresowych znajdujące się w prawym panelu programu. W oknie zostają umieszczone wszystkie etykiety występujące w programie wraz z przypisanymi do nich adresami. Lista etykiet ułatwia zlokalizowanie fragmentu kodu w buforze.



#### Kompilacja

(Menu -> Narzędzia -> Kompiluj / Pasek narzędziowy: tak / Skrót: F9)

Funkcja uruchamia proces kompilacji. Po zakończeniu kompilacji znacznik zajętości bufora jest aktualizowany i wskazuje ilość bajtów kodu wynikowego w buforze.

#### Śledzenie etykiet adresowych

(Menu -> Narzędzia -> Ustawienia / Pasek narzędziowy: nie)



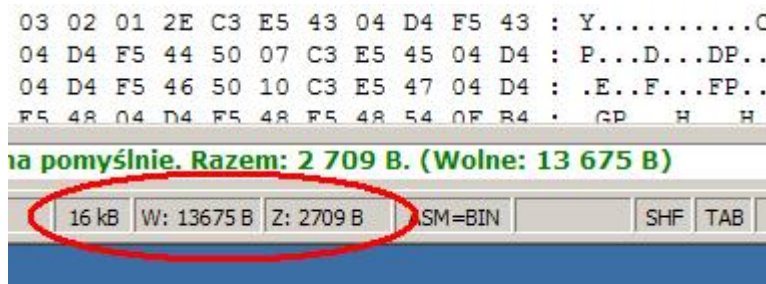
Po włączeniu tej funkcji, jeśli okno etykiet jest wypełnione, każde kliknięcie na etykiecie w tym oknie automatycznie odszukuje odpowiadający fragment kodu źródłowego w oknie edytora i podświetla go. Dodatkowo, jeśli otwarte jest okno podglądu bufora, również tutaj podświetlany jest odpowiadający wiersz kodu wynikowego.

#### 4. BUFOR KODU

Bufor kodu wynikowego ma maksymalną pojemność 16 kB i jest dopasowywany automatycznie do pojemności wybranego układu (rozmiar obszaru roboczego ulega zmniejszeniu do wartości odpowiadającej pojemności układu wybranego do programowania). Gdy program nie obsługuje żadnego sprzętu lub nie wybrano żadnego układu dostępny jest bufor o maksymalnej pojemności.

Jeśli nastąpi przekroczenie wybranej (w sposób automatyczny) pojemności bufora kompilator zasygnalizuje błąd.

Aktualną pojemność bufora (w kB), obszar zajęty i wolny (w bajtach) wskazuje znacznik bufora na pasku statusu:



gdzie:

- **16 kB** oznacza całkowity obszar roboczy bufora (pojemność bufora)
- **W: 13675 B** – wolna przestrzeń bufora (znacznik obszaru wolnego)
- **Z: 2709 B** – zajęta przestrzeń bufora (znacznik zajętości bufora)

#### Czyść bufor

(Menu -> Bufor -> Czyść bufor / Pasek narzędziowy: nie)

Funkcja umożliwia wypełnienie całej przestrzeni bufora wartościami FFH.

#### Ładuj bufor

(Menu -> Bufor -> Ładuj bufor / Pasek narzędziowy: tak)

Ładuje do bufora zawartość wskazanego na dysku pliku. Plik może mieć format hexadecymalny (IntelHEX) lub binarny. Po załadowaniu znacznik zajętości bufora wskazuje ilość załadowanych bajtów.

#### Zapisz bufor

(Menu -> Bufor -> Zapisz bufor / Pasek narzędziowy: tak)

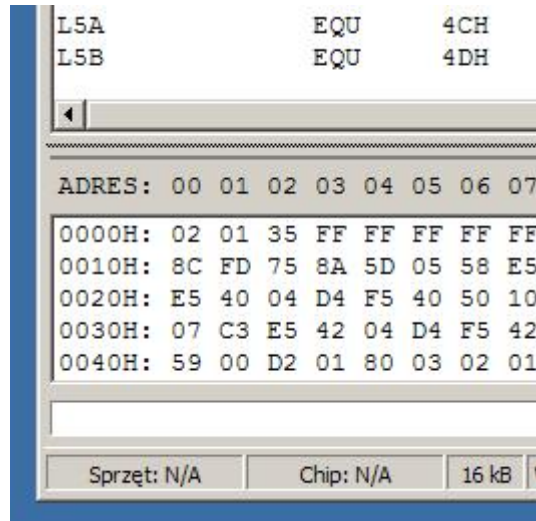
Zapisuje do nowego / wskazanego na dysku pliku zawartość bufora. Plik może mieć format hexadecymalny (IntelHEX) lub binarny. Zapisywana jest tylko taka część, jaką wskazuje znacznik zajętości bufora.

#### Podgląd bufora

(Menu -> Bufor -> Podgląd bufora / Pasek narzędziowy: tak / Skrót: F12)

Ta funkcja otwiera / ukrywa w dolnej części okna edytora dodatkowy okno, w którym można przeglądać zawartość bufora kodu wynikowego wraz z adresem przypisanym do każdego wiersza oraz offsetem dla każdej kolumny.

Rozmiar okna bufora można zmieniać poprzez przeciąganie ukrytej krawędzi w jego nagłówku nieco ponad linią separacyjną (na poniższym rysunku linia kropkowana):



## 5. OBSŁUGA SPRZĘTU

Jeśli do komputera podłączono właściwy sprzęt, program automatycznie wykryje go i uaktywni funkcję „Obsługa sprzętu”. Po jej wybraniu otworzy się okno, w którym można skonfigurować sprzęt a także wykonać każdą z możliwych operacji na wybranym programatorze i chipie.

### Obsługa sprzętu

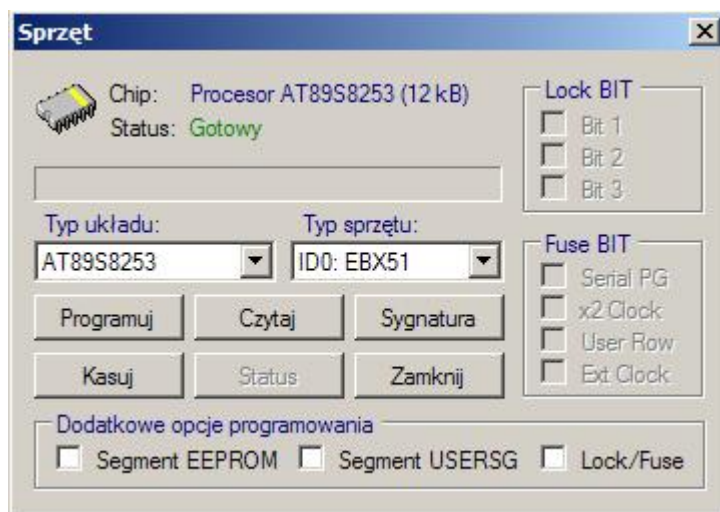
(Menu -> Narzędzia -> Sprzęt ... / Pasek narzędziowy: tak)

W otwartym oknie obsługi sprzętu można przeprowadzać następujące operacje:

- Wybór urządzenia (programatora)
- Wybór układu do programowania (chip)
- Kasowanie
- Programowanie
- Odczyt zawartości
- Programowanie bitów zabezpieczających i konfiguracyjnych
- Odczyt bitów zabezpieczających i konfiguracyjnych
- Odczyt sygnatury

Ponadto dla procesora AT89S8253 można wybrać:

- Programowanie i odczyt wewnętrznej pamięci EEPROM 2kB
- Programowanie i odczyt sygnatury użytkownika 64 B



Funkcje:

### Programuj

Programuje zawartość wybranego układu zawartością bufora programu. Programowana jest cała przestrzeń wybranego układu. Przed rozpoczęciem procedury programowania zawartość układu jest automatycznie kasowana.

### Kasuj

Kasuje zawartość układu

### Czytaj

Odczytuje zawartość układu i przesyła ją do bufora programu.

### Status

Odczytuje stan bitów zabezpieczających i konfiguracyjnych. Funkcja dostępna po uaktywnieniu opcji „Lock/Fuse” w sekcji „Dodatkowe opcje programowania” i tylko wtedy, kiedy dany układ obsługuje taki odczyt.

### Sygnatura

Odczytuje sygnaturę układu i jeśli odczyt jest poprawny wybiera automatycznie odpowiedni typ układu z listy.

### Dodatkowe opcje programowania

Można tutaj wybrać możliwość programowania dodatkowych segmentów, jeśli takowe posiada wybrany typ układu. Jeśli zaznaczono odpowiedni segment zostanie on użyty w czasie programowania układu funkcją **Programuj**.

Dodatkowo w tej sekcji znajduje się bezpiecznik **Fuse/Lock** odblokowujący dwie sekcje odpowiedzialne z programowanie bitów zabezpieczających i konfiguracyjnych.

### Lock BIT

Sekcja, w której można ustawić bity zabezpieczające programowaną zawartość. Jeśli jest aktywna zostanie użyta w czasie programowania układu. Jeśli wybrany układ obsługuje taką możliwość, można odczytać stan bitów funkcją **Status**.

### Fuse BIT

Sekcja odpowiadająca za bity konfiguracyjne wybranego układu. Jeśli jest aktywna zostanie użyta w czasie programowania układu. Stan bitów można odczytać funkcją **Status**.

## 6. SZYBKIE PROGRAMOWANIE

Ta funkcja dostępna jest tylko w przypadku podłączenia właściwego sprzętu oraz wybrania układu do programowania. Umożliwia zaprogramowanie wybranego układu tylko takim obszarem bufora, jaki został wykorzystany podczas kompilacji lub ładowania bufora z pliku. Ilość przesłanych danych nie musi się pokrywać ze znacznikiem zajętości bufora, ponieważ program przesyła dane w 64 bajtowych paczkach. Przesłanych zostanie tyle paczek ile jest wymaganych do zaprogramowania układu zajęтым obszarem bufora.

Funkcja bardzo przydatna na etapie dopracowywania kodu, umożliwia szybkie zaprogramowanie układu i natychmiastowe sprawdzenie pracy programu w docelowym środowisku (w przypadku zastosowania płyty uruchomieniowej EBX51).

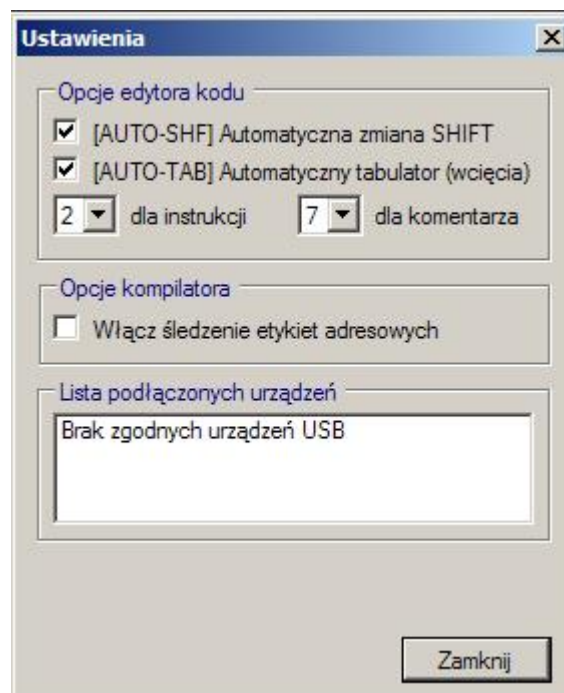
### Szybkie programowanie

(Menu -> Narzędzia -> Programuj ... / Pasek narzędziowy: tak / Skrót: F11)



## 7. USTAWIENIA

W tym oknie dostępne są globalne ustawienia programu ASMX51. Po otwarciu okna można wybrać następujące opcje konfiguracyjne:



**AUTO-SHF**

Automatyczna zmiana wielkości znaków edytora (opisana w sekcji **Edytor**)

**AUTO-TAB**

Automatyczne wstawianie tabulatorów (opisana w sekcji **Edytor**). Liczbę tabulatorów dla instrukcji i komentarza można ustawić niezależnie.

**Śledzenie etykiet adresowych**

Po włączeniu tej opcji możliwe jest sprawdzanie lokalizacji wybranej etykiety w kodzie źródłowym i w buforze (opisana w sekcji **Edytor**).

**Lista podłączonych urządzeń**

Lista wyświetla wszystkie podłączone i zgodne z programem urządzenia. Urządzenia z tej listy dostępne są w oknie **Obsługa sprzętu** na liście rozwijanej **Typ sprzętu**.

**8. LICENCJA**

Niniejsza Licencja uprawnia do bezpłatnego korzystania z oprogramowania "ASMX51".

**Uwaga!**

Należy dokładnie zapoznać się z treścią niniejszej Umowy Licencyjnej, zwanej dalej Licencją, która stanowi prawnie wiążącą umowę zawartą pomiędzy osobą fizyczną lub prawną, zwaną dalej Użytkownikiem a producentem tego programu zwanym dalej Producentem.

Oprogramowanie chronione jest polskim i międzynarodowym prawem autorskim i o ochronie praw własności intelektualnej.

1. Przedmiotem niniejszej Licencji jest użytkowanie oprogramowania "ASMX51" zwanego dalej Programem.
2. Producent jest wyłącznym właścicielem praw autorskich do Programu.
3. Producent przekazuje bezpłatnie kopię Programu Użytkownikowi do użytku na dowolnie wybranej ilości stanowisk komputerowych.
4. Użytkownik ma prawo udostępniać dowolną liczbę kopii Programu osobom trzecim wyłącznie nieodpłatnie.
5. Zabroniona jest dekompilacja, deasemblacja oraz jakakolwiek inna modyfikacja Programu, a także wykorzystanie Programu lub jego części w innym oprogramowaniu.
6. Dozwolone jest wykorzystanie Programu w publikacjach bez odrębnej zgody Producenta z zastrzeżeniem umieszczenia czytelnej informacji o Producentcie Programu.
7. Producent nie ponosi żadnej odpowiedzialności za powstałe szkody, które mogłyby nastąpić na skutek stosowania Programu, w tym również utratę informacji i straty pieniężne wynikłe z użytkowania lub niemożności użytkowania Programu.
8. Producent nie ponosi żadnej odpowiedzialności wobec niezgodnego z prawem korzystania z Programu.
9. W przypadku naruszenia postanowień Licencji, Producent ma prawo rozwiązać niniejszą umowę oraz dochodzić swoich roszczeń na drodze prawnej.

Producent: DK © 2012 Wszystkie prawa zastrzeżone.