

**MTU8B5X**

**8-Bit Micro-Controller**

**Cross assembler and Writer**

**User's Manual**





## **MTU8BX Cross Assembler and Writer Contents**

---

	Page
1. PREFACE .....	3
2. SYSTEM CONFIGURATION.....	3
3. INSTALLATION .....	4
4. OPERATIONS .....	5
4-1. File menu .....	6
4-2. Edit menu .....	9
4-3. Code menu .....	9
4-4. Compile menu .....	12
4-5. Tools menu .....	13
4-6. Settings menu .....	14
4-7. Window menu .....	16
4-8. About menu .....	17
5. MDI CHILD WINDOW .....	18
6. OPERATION FLOW .....	20

### **1. PREFACE**

8B5X is a high-performance cross assembler and writer for developing the application systems of the JA5854E, JA5855E, JA5856E and JA5857E. It has many features and enhancements:

1. Use MDI ( Multiple Document Interface ) interface, Users could operating between differences windows
2. In compiling-time, Cross Assembler could check the case sensitive of labels and compile with specific parameters. The sequence of DATA and CODE sections has no more restriction when compiling.
3. After compilation, users could write code into the chip of MTU8B5X Writer Board and verify it. Users could convert .OBJ ( PIC can choose .HEX ) code between JA5854E/55E/56E/57E and PIC16C54/55/56/57.

### **2. SYSTEM CONFIGURATION**

The package of MTU8B5X cross assembler and writer contains a diskette with development software and a PCB boards ( Parallel (Printer) port interface ).

#### **2-1. Hardware**

- MTU8B56/57 Writer Board:

#### **2-2. Software**

- 8B5X.exe : Main program.
- ASM.ini : Initialize file.
- Other related files.

#### **2-3. PC System**

- IBM PC 486/586 or compatible machines.
- Windows 95 or later.

### 3. INSTALLATION

#### 3-1. Hardware

1. Plug the connector of the power adapter ( 15V~20V ) into the power connector on the MTU8B5X Writer Board.
2. Connect the parallel port of PC and parallel connector on the MTU8B5X writer board.
3. To confirm parallel port is EPP mode.

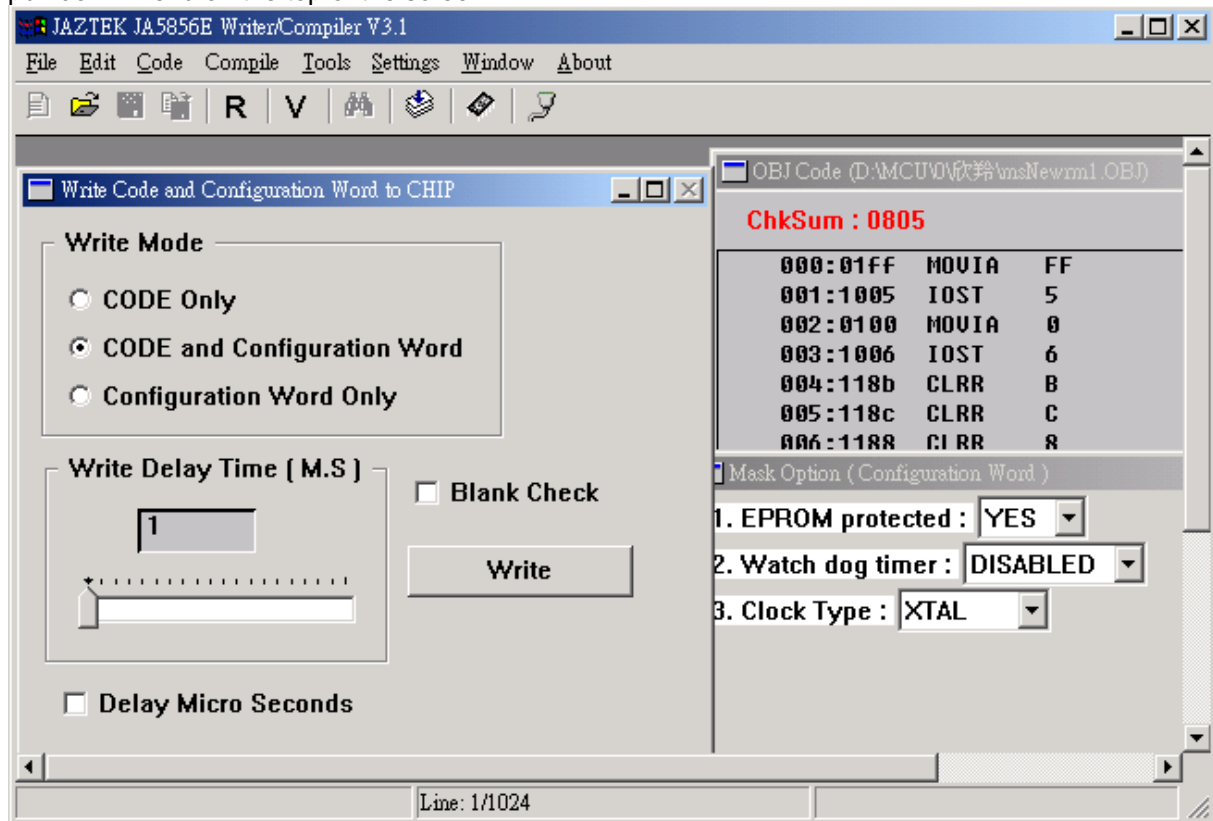
**Caution:** The VDD(+) is *inside* the power connector.

#### 3-2. Software

1. Please install the "port95nt.exe" before setup JAuP8.  
<http://www.driverlinx.com/DownLoad/DIPortIO.htm>
2. Run JAuP8-Vx-Setup.exe to install programs.

#### 4. OPERATIONS

After run JAUP8 , the main window will be shown on the screen. This main window includes 8 pull-down menu on the top of the screen.



In this window, users can use mouse or press [Alt+F], [Alt+E], [Alt+C], [Alt+P], [Alt+S], [Alt+W], [Alt+A] keys ) to activate the pull-down menu. In the pull-down menu, users can move the mouse cursor or ↑, ↓ keys to move the highlight bar to each item and then press [Enter] key or press mouse left button to execute the function.

Have a toolbar where on the manu bar under, users can move the mouse cursor to the button of toolbar, then press the mouse left button to execute the function

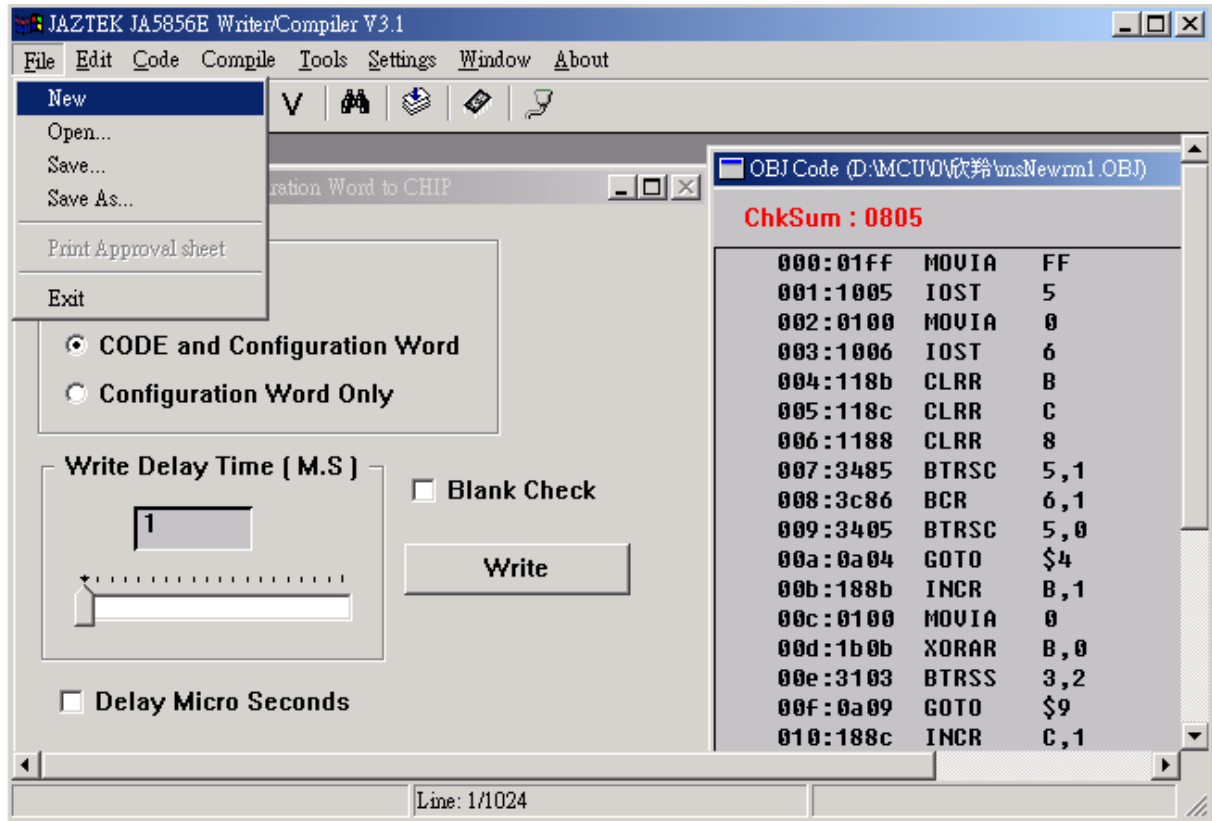
In this main window, it contains three child windows which are :

1. **Write Code and Configuration Word** window : Users must choose **write mode** and set the **delay time** ( MilliSecond ) when write codes to the chip.
2. **OBJ Code** window : This is a memory buffer of OBJ code, users can save OBJ code from or load code to the window, you can press enter key or double click the mouse left button button to modify the code data ( Input as Mnemonic type, Ex : sleep ), if decide to update please press "Enter" key, otherwise press "Esc" key.
3. **Mask Option** window : Users can choose option by the window.

## 4-1. File menu

There are five functions in this sub-menu:

1. **New**
2. **Open...**
3. **Save...**
4. **Save As...**
5. **Exit**



### 4-1-1. New

This function will fill NOP code in **OBJ Code** windows.

#### Operation:

Choose the **"New"** item from the sub-menu, it will create a blank OBJ code ( fill all codes as NOP )

#### Limitation:

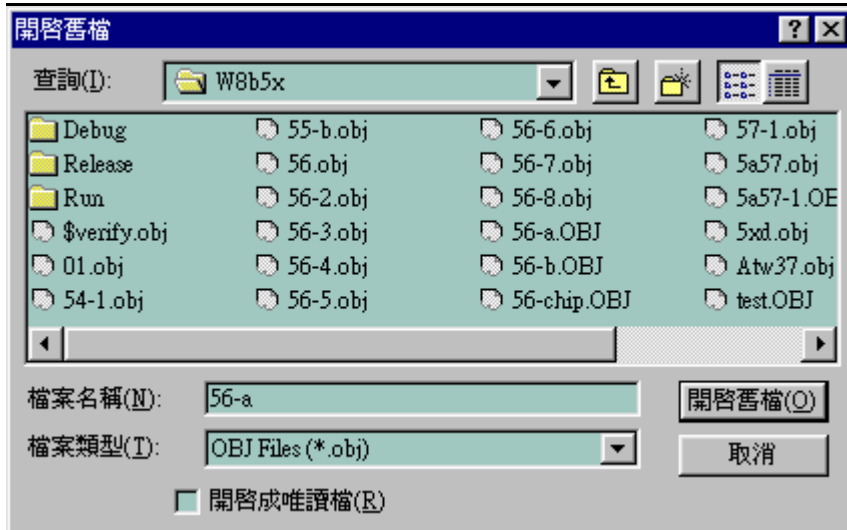
The active window must be location at **OBJ Code** window.

### 4-1-2. Open...

This function will open exists .OBJ files. ( Ex: Test.OBJ ).

#### Operation:

Choose the **"Open..."** item from the sub-menu, the screen will shown as below :



Please input exists file name to open, the extension filename is .OBJ, if open success, the content of .OBJ and will shown on “**OBJ Code**” window and “**Mask Option**” window , otherwise, an error message will display on screen.

Limitation:

None.

#### 4-1-3. Save

The function will save .OBJ code data and configuration word data as exists file name.

Operation:

First, please choose the focus to an active child window ( **OBJ Code** window ), after choose the “**Save...**” item from the sub-menu, it will save data to file as now

Limitation:

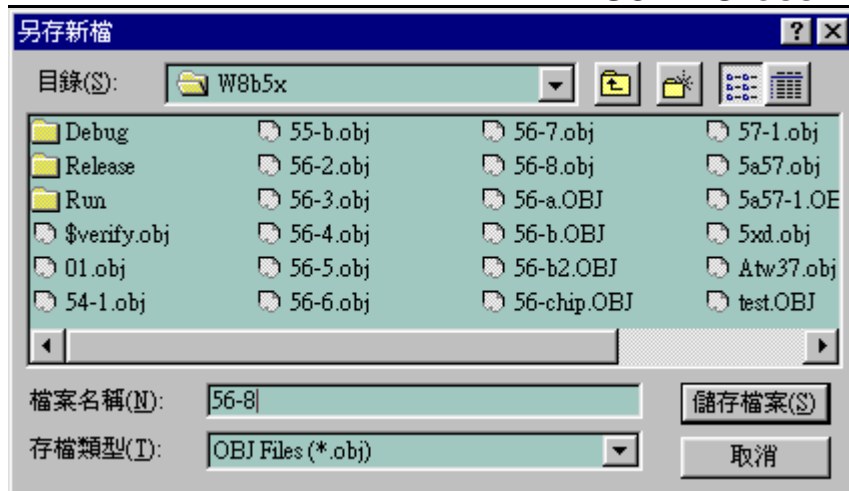
The active window must be location at **OBJ Code** window, otherwise this function will be disabled..

#### 4-1-4. Save As

The function will save .OBJ code data and configuration word data as input file name.

Operation:

First, please setting focus to an active window ( OBJ Code window ) after choose the “**Save As...**” item from the sub-menu, the screen will display a dialog box and allows to input file name, the following shows how the save dialog box appears:



Limitation:

The active window must be location on **OBJ Code** window.

**4-1-5. Exit**

The function will close and exit the window of MTU8B5X writer/compiler.

Operation:

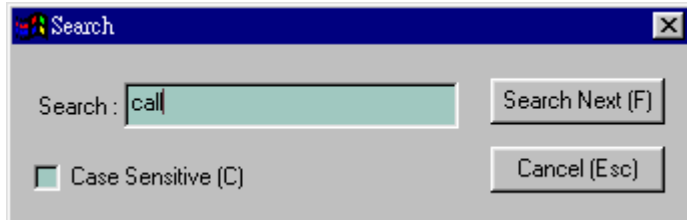
choose the **"Exit"** item from the sub-menu, it will close and exit the window

Limitation:

None.

## 4-2. Edit menu

This is a virtual ( unavailing ) menu item while active window is location on “**Write Code**” window or “**Mask Option**” window, otherwise it will contain a “**Search**” function in the sub menu, if selection it,

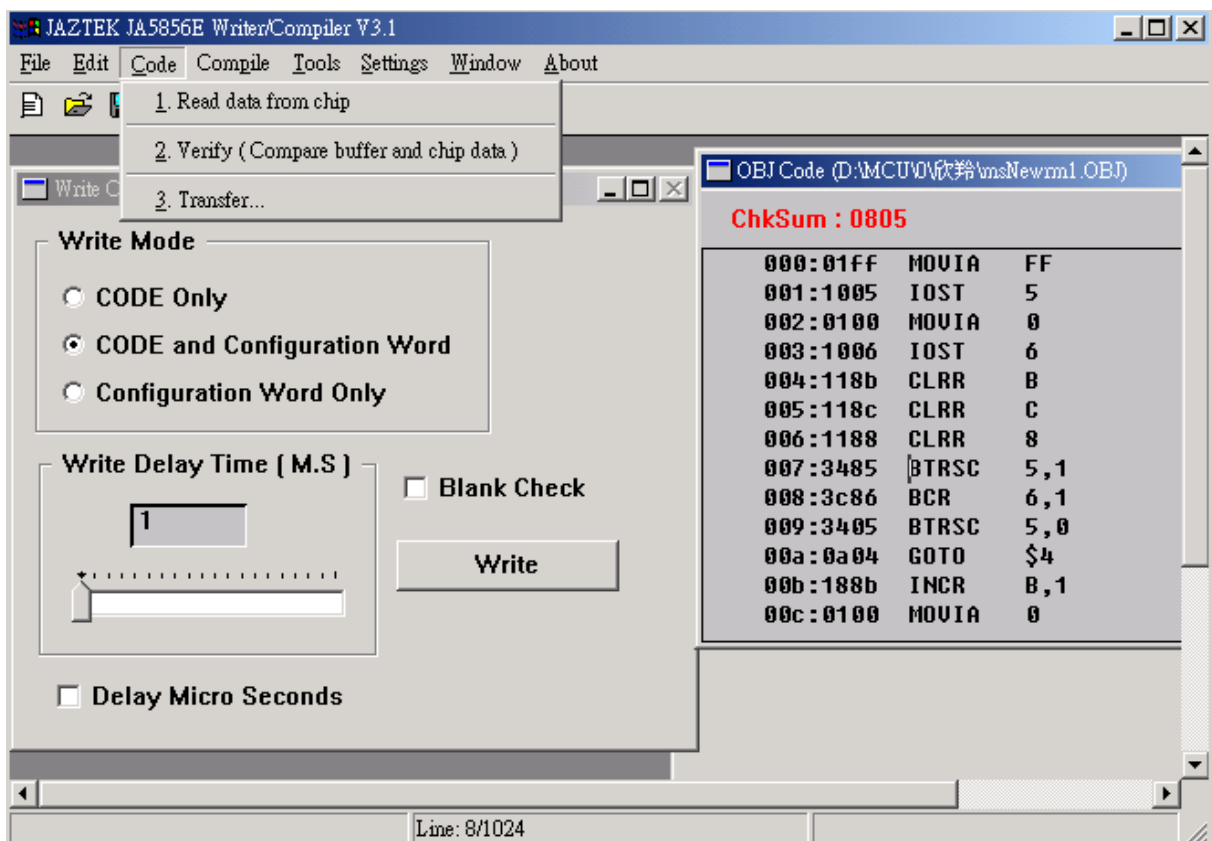


the screen will shown as below :  
You can type a string to search by the dialog box function.

## 4-3. Code menu

There are three functions in this sub-menu:

1. **Read data from chip**
2. **Verify ( Compare buffer and chip data )**
3. **Transfer...**



### 4-3-1. Read data from chip

This function will read the CHIP data back from the MTU8B5X writer board, after load data into memory buffers and displays it on **OBJ Code** window.

#### Operation:

Choose the “**Read data from chip**” item from the sub-menu, the system will read data from CHIP.

If Chip data was blank or on protected mode, a message will prompted on the screen and stop to load data into memory buffers.

Limitation:

If the writer board is not in the ready state, some error messages will be prompted on the screen.

### 4-3-2. Verify ( Compare buffer and chip data )

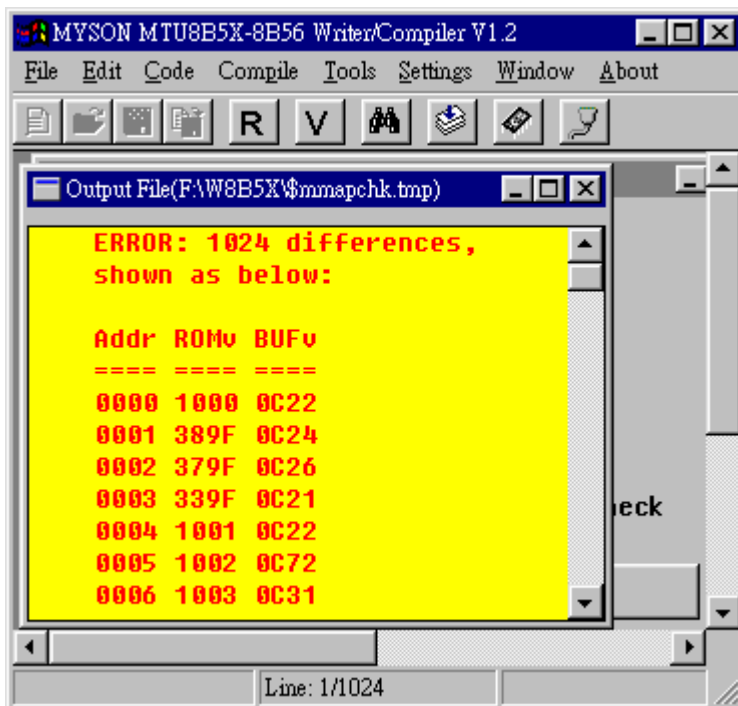
This function will compare of the OBJ Code memory buffers and Chip data.

Operation:

Choose the “**Verify**” item from the sub-menu, the system will compare of the OBJ code memory



buffer and chip data, if no differences be found, the screen will displays:



Otherwise, the screen will displays differences encounter as below :

Limitation:

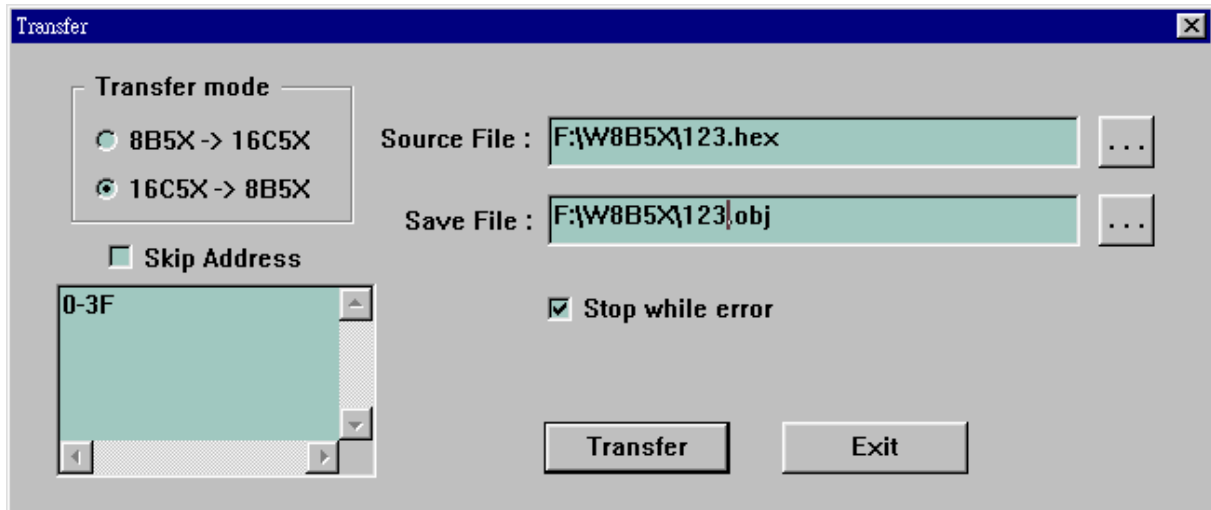
If the system is not in the ready state, some error messages will be prompted on the screen.

### 4-3-3. Transfer...

This function will convert code from MTU8B54/55/56/57 to PIC16C54/55/56/57 and oppositely from PIC16C54/55/56/57 to MTU8B54/55/56/57.

#### Operation:

Choose the “*Transfer...*” item from the sub-menu, the screen will displays a dialog box as below :



First, please choose **transfer mode** and input source file and save file from above dialog box then press “Transfer” button begin to convert code.

If has errors occurred when convert code, some error messages will be prompted on the screen

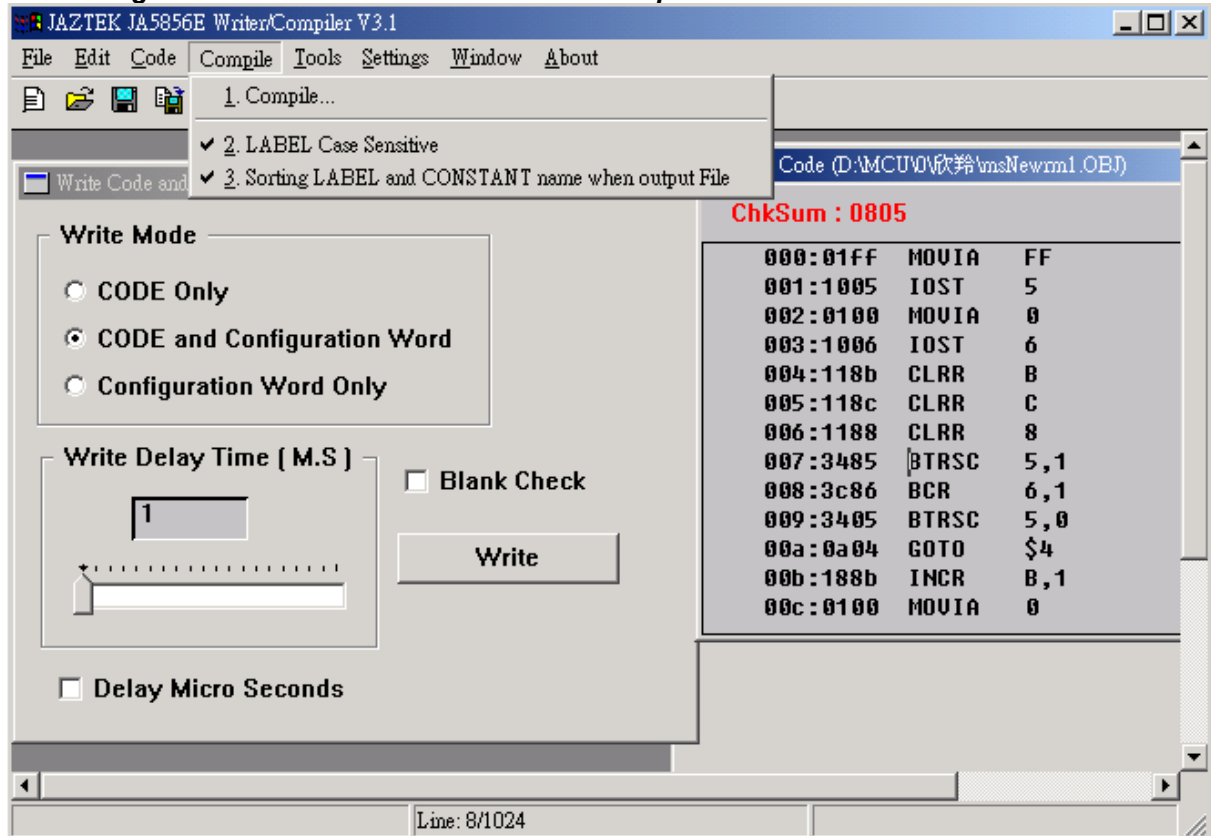
#### Limitation:

None

### 4-4. Compile Menu

There are one functions and two option in this sub-menu:

1. **Compile**
2. **LABEL Case Sensitive**
3. **Sorting LABEL and CONSTANT name when output File**



#### 4-4-1. Compile...

This function will compile your .ASM file and create some output files when end of compile.

##### Operation:

Choose "**Compile...**" item from the sub-menu. The Open File" dialog box will be displayed. Select the desired .ASM file to compile. During compilation some status information will be shown on the screen:

- (1) If errors occur after compile, the output files .ERR and .ASM will displays on the screen, if you requires the more detail of messages, please refer to .LST file.
- (2) The LABEL name will output to .LBL file, the CONSTANT name will output to .EQU file.
- (3) If no error occurs, the system will create an OBJ file and the screen will display: "**Cross Assemble OK.**".

##### Limitation:

None.

#### 4-4-2. LABEL Case Sensitive

This option could enabled/disabled the case sensitivity in compiling-time. If choose "enabled", the compiler will automatically check case of labels. So that Sum, sum and suM are distinct identifiers.

##### Operation:

Choose the **"LABEL case check"** item from the sub-menu to select/unselect this function. This setting will be saved into the initial file.

Limitation:

None

**4-4-3. Sorting LABEL and CONSTANT name when output File**

This option could enabled/disabled sorting label and constant name in the output file. If select "enabled", the .LBL and .EQU file will list label and constant name alphabetically.

Operation:

Choose the **" Sorting LABEL and CONSTANT name when output File"** item from the sub-menu to enable/disable this function. This setting will be saved into the initial file.

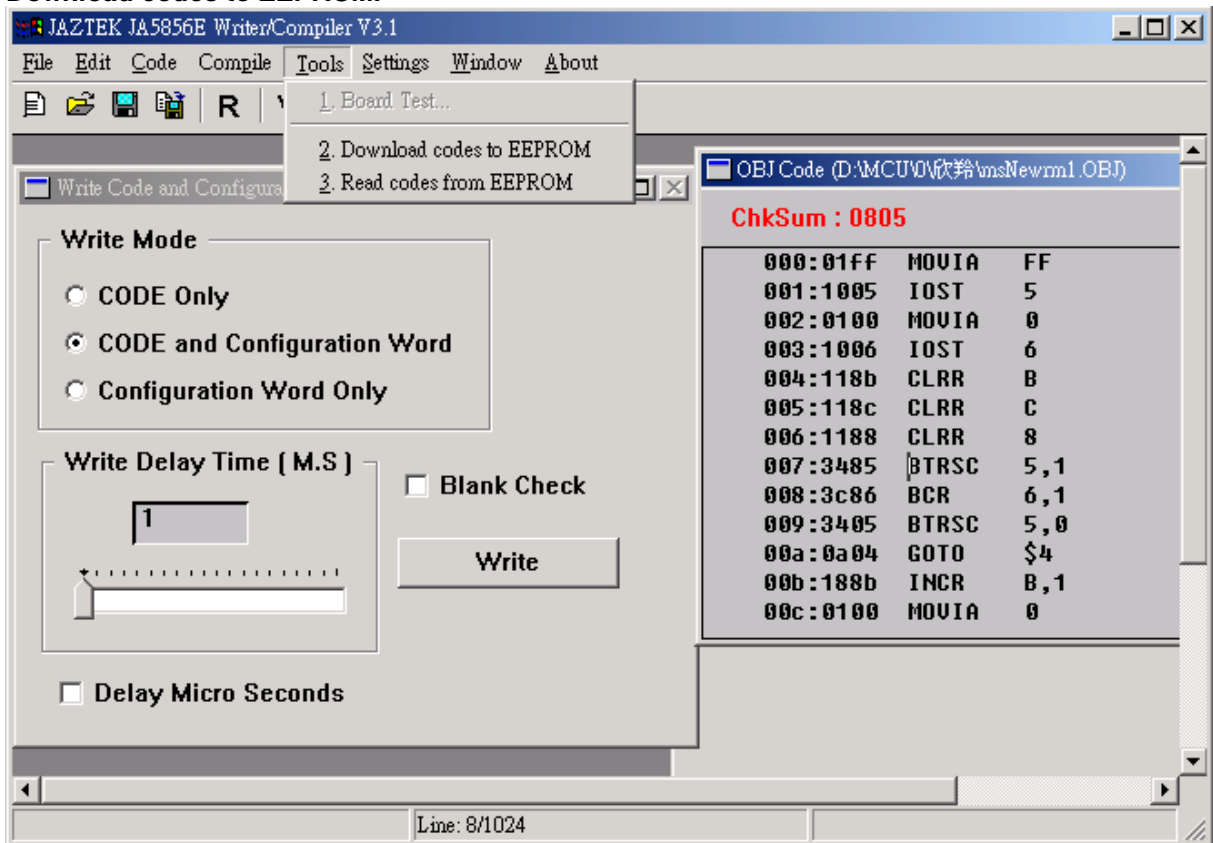
Limitation:

None

### 4-5. Tools menu

There are three options in this sub-menu:

1. **Board test...**
2. **Download codes to EEPROM.**



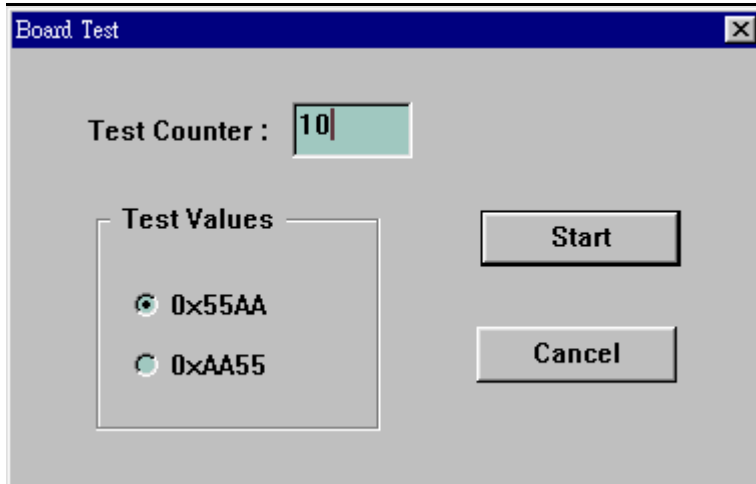
3. **Read code from EEPROM.**

#### 4-5-1. Board Test...

The function will testing the writer board is OK or NO.

Operation:

Choose the **"Board Test..."** item from the sub-menu, the screen will shown a dialog box as below:



First, please settings the “**Test Counter**” value ( 1~999 ) and choose the “**Test Values**” (0x55AA or 0xAA55 ), then move the mouse cursor to “**Start**” button and push the left button start to testing.

Limitaton : None

#### **4-5-2. Download codes to EEPROM.**

The function will write the OBJ code buffer data to EEPROM ( not MTU8B5X ).

Operation:

Choose the “**Download codes to EEPROM**” item from the sub-menu, the system will write the .OBJ codes buffer data into the EEPROM, if have error occurred, an error message will displays on the screen.

Limitation : None

#### **4-5-3. Read codes from EEPROM.**

The function will read the data from the EEPROM into the .OBJ memory buffers.

Operation:

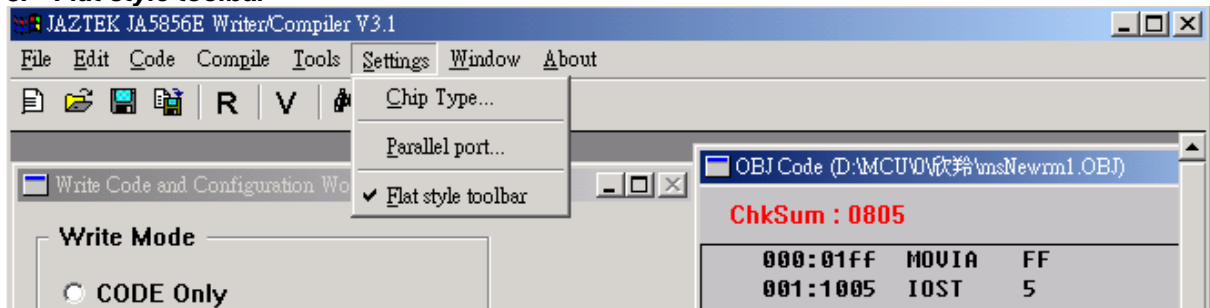
Choose the “**Read codes from EEPROM**” item from the sub-menu, the system will read the data from the EEPROM into the memory buffers and shown the code data on the the screen.

Limitation : None

## 4-6. Setting menu

There are three options in this sub-menu:

1. **Chip Type**
2. **Parallel Port**
3. **Flat style toolbar**



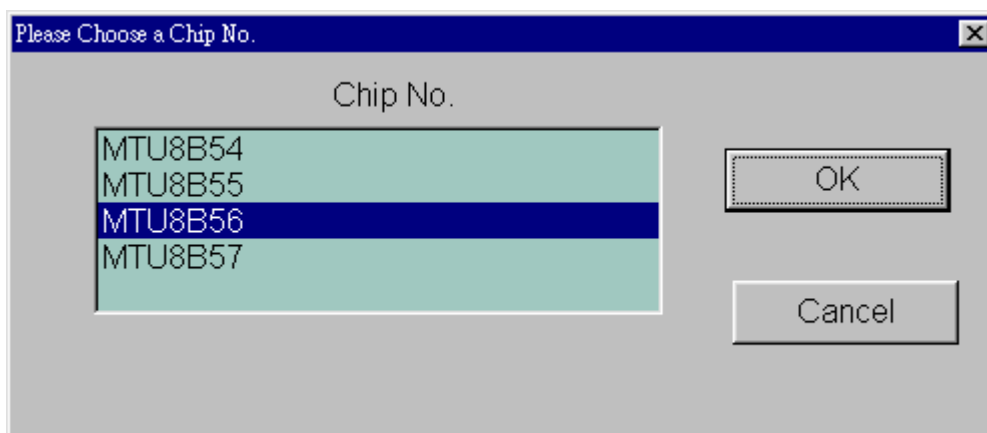
Use these options to specify chip type and environment settings. Some settings will be saved into the initial file.

### 4-6-1. Chip Type

This option could choose the desired chip type which are : **MTU 8B54**, **MTU8B55**, **MTU8B56** or **MTU 8B57**. You must be choice the chip type before when you want to process the compile, transfer or read/write the data...and so on.

#### Operation:

Choose the **“Chip Type”** item from the sub-menu, the screen will shown a dialog box as below:



Use **↑**, **↓** keys to move the highlight bar to the destination and then press **[Enter]** key or double click the mouse left button to confirm this setting. This setting will be saved into the initial file. When you choose differ

#### Limitation:

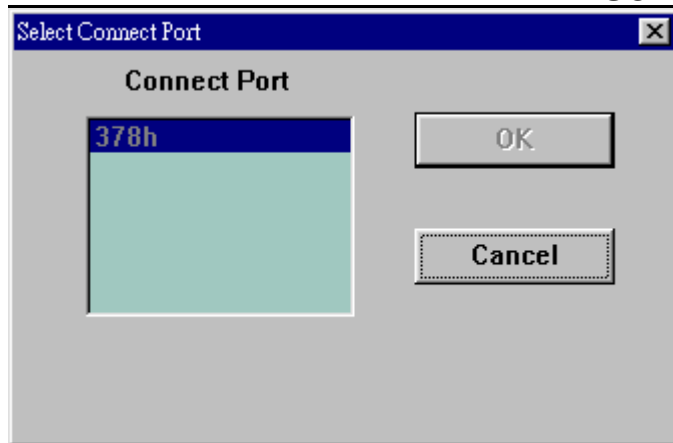
None.

### 4-6-2. Parallel port

This option could choose the parallel port parament.

#### Operation:

Choose the **“Parallel port”** item from the sub-menu, a dialog box will be displayed:



Use **↑**, **↓** keys to move the highlight bar to the destination and then press **[Enter]** key or double click the mouse left button to confirm this setting. This setting will be saved into the initial file.

Limitation:

If only one parallel port be detected , then it will be set as default port, the function will disabled.

#### **4-6-3. Flat style toolbar**

This option could choose the display style of the tool bar.

Operation

Choose the "**Flat style toolbar**" item from the sub-menu to select/unselect this function. If the "**Flat style toolbar**" is selected, the toolbar will displays as flat mode, otherwise it will display as button mode, this setting will be saved into the initial file.

Limitation:

None

#### **4-7. Window menu**

The function could choose the display type and set the active windows of MDI child window, the MDI window display type will shown as below:

**4-7-1. Horizontal type** : Tiles MDI child windows so that they are wide rather than tall.

**4-7-2. Vertical type** : Tiles MDI child windows so that they are tall rather than wide.

**4-7-3. Cascade type** : Arrange all MDI child windows in a cascade format.

**Limitation** : Only can minimum to an icon size of "**Write Code**" window ( can't resize ),



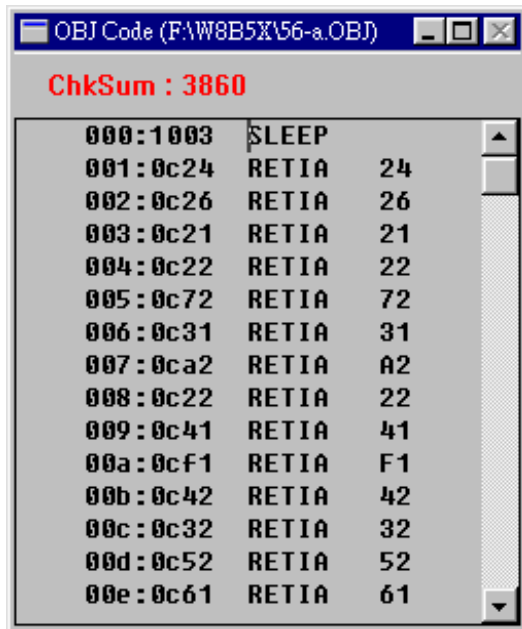
**4-8. About menu**

This function will display the version number of the MTU8B5X Writer/Compiler and the corresponding information.

### 5. MDI CHILD WINDOW

There are three MDI windows is residing at client area of main window, they are :

**5-1. OBJ Code window :**



**This window will shown the content of OBJ code, it contains following 3 items:**

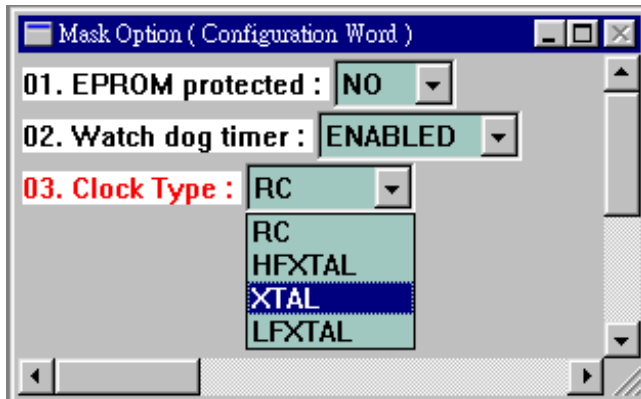
- (1). PC address.
- (2). Machine Code.
- (3). Mnemonic Code.

You can view or to scroll the data by use the up,down arrow key or Page Up,Page Down key or to move the mouse cursor to the scroll bar where the right of window then pressing the mouse left button.

If you want to modify the OBJ code in the window, please press "Enter" key or double click the mouse left button while the caret's current position is location on mnemonic code above, then input mnemonic code and press "Enter" key to modify, press "Esc" key to cancel modified.

The OBJ data can be saved to file, a message box will prompted if save file already exists or difference between the OBJ code memory values and the exists file values.

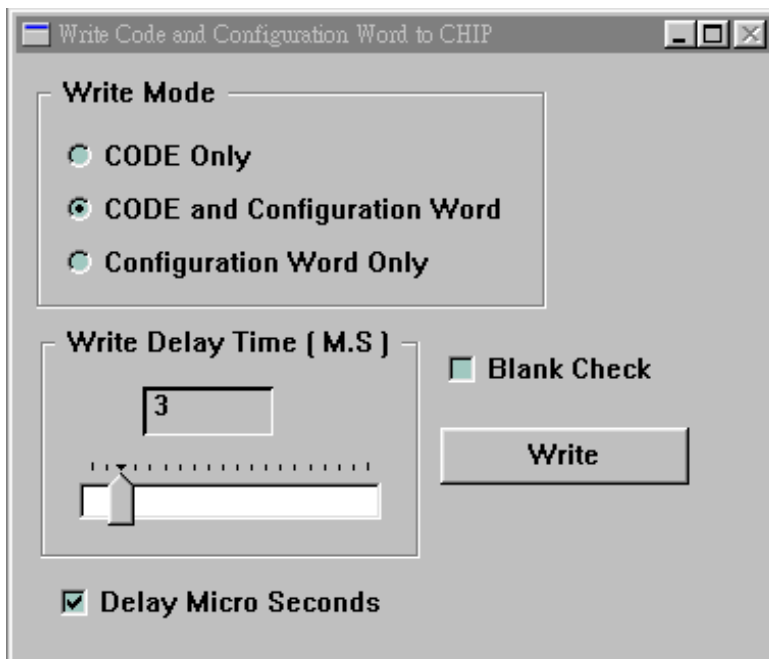
**5-2. Mask Option window:**



This window can choose the MASK option or OTP option of chip, please move the mouse and use its left button to choose the options, the option contains following 3 items:

- (01). EPROM protected : Choose “NO” or “YES”.
- (02). Watch dog timer : Choose “ENABLED” or “DISABLED”.
- (03). Clock Type : Choose “RC” or “XFXTAL” or “XTAL” or “LFXTAL” modes.

The option data will be load from or save to OBJ file when open or save OBJ codes data.



**5-3. Write Code and Configuration Word to CHIP window**

This window can specifies the writing mode and setting the delay time when write the OBJ code into chip.

The write mode contains three types, they are:

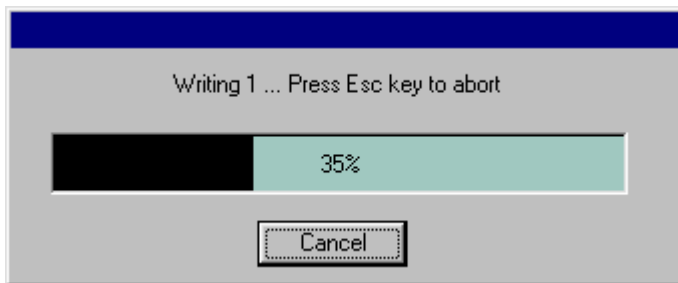
- (1). **Code Only** : Only write the memory buffers of the OBJ codes into the chip.
- (2). **Code and Configuration Word** : Write the memory buffers of the OBJ codes

and Configuration Word ( Option ) into the chip.

(3). **Configuration Word** : Only write the Configuration Word into the chip.

You can specify the writing delay time ( for every address, in milliseconds or microseconds) when writing the OBJ code into the chip. the value of delay time is limited from 1 to 20 milliseconds ( propose value is from 3 to 6 depend on type of chip package ) while the “**Delay Micro Seconds**” is *unselected*, if “**Delay Micro Seconds**” is *selected*, the write delay time of every address is 500 micro seconds ( 0.5 millisecond ).

If you want to write the OBJ codes to the chip, please move the mouse cursor to the position of “**Write**” button and then pressing the left button to execute it, after the screen will displays a **meter window** as below ( choose (1) or (2) ):



\*\*\* If the write delay time is less than 3 seconds, the meter window will no displays.

The function will continue to fills the meter window by use the system color until codes writing finished, you can press “Esc” key or press “**Cancel**” button to break the writing.

When the OBJ code is writing finished, the system will read the data from the chip automatically and compare of the OBJ code memory data, if no differences encounter, the screen ( Under **Write** button ) will display “**Success !**” message, otherwise the it will display “**Failure !**” message and create a error message window to displays the chip data and OBJ code data.

You can selects or clears the “**Blank Check**” control button to checked/unchecked the blank state of chip data, if the button is checked, before the system will read the data from chip when writing code and the check it is in blank state or not, if chip is not blank, a warning message will shown on screen.

## 6. OPERATION FLOW

### 6-1. How to Write Code To Chip ?

The following is the step for how to write code to chip :

- (1). Specifying the “Chip Type” from manu bar or tool bar.
- (2). Choose “Open File ” to open the “.OBJ” file from the menu bar or tool bar and load it into the “OBJ Code” MDI window.
- (3). Specifying the configuration word ( option ) values from the “**Mask Option**” MDI window.
- (4). Specifying the “Write Mode” and “Write Delay Time “ from the “**Write Code and Configuration Word to CHIP**” MDI window.
- (5). Checking the writer board and IC chip ( MTU8B5X ) is in ready state.
- (6). Pressing “Write” button from the “**Write Code and Configuration Word to CHIP**” MDI window to write codes and waiting for until finished, the window will shown



“Success !” if writing is successful, otherwise it will shown “Failure ! ”.

**6-2. How to verify code ?**

- (1). Specifying the “Chip Type” from manu bar or tool bar.
- (2). Choose “Open File ” to open the “.OBJ” file from the menu bar or tool bar and load it into the “OBJ Code” MDI window.
- (3). Execute the “Verify” function from menu bar or tool bar, the system will read data from chip and compare of OBJ code memory buffers after the screen will displays the results of verify.