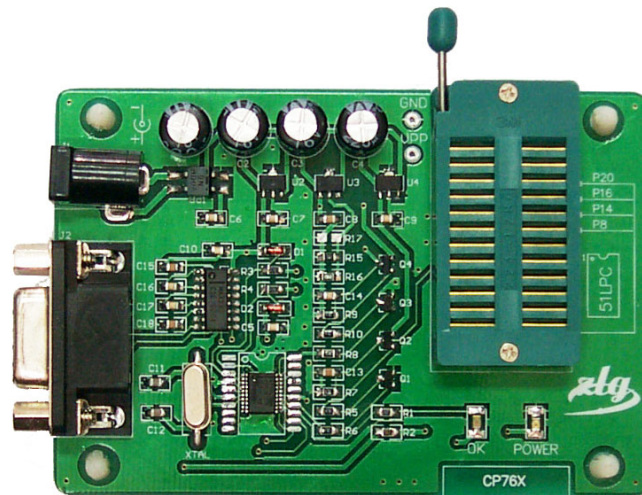


CP76X USER MANUAL



Guangzhou ZHIYUAN Electronic Co., LTD.

2002 August 23

DIRECTORY

- 1 General Description**
- 2 Setup**
- 3 Main Window**
- 4 Menu Function**
- 5 Function Description**
- 6 Settings**

Appendix 1 Store Configuration Information in Hex File

Appendix 2 An Example of Programming 51LPC Series MCUs with CP76x

1. GENERAL DESCRIPTION

CP76x programmer is used to program 51LPC series MCUs.

FEATURES

- High Reliability
- High Speed
- Comfortable Operation
- Device auto detection
- Supports 51LPC full series MCUs
- Blank Check and verification function
- Serial number auto increment is available

Device Types

P87LPC759 1K OTP MCU
P87LPC760 1K OTP MCU
P87LPC761 2K OTP MCU
P87LPC762 2K OTP MCU
P87LPC764 4K OTP MCU
P87LPC767 4K OTP MCU
P87LPC768 4K OTP MCU
P87LPC769 4K OTP MCU
P87LPC776 8K OTP MCU
P87LPC777 8K OTP MCU
P87LPC778 8K OTP MCU
P87LPC779 8K OTP MCU

PARTS LIST

1. CP76x PROGRAMMER	1 piece
2. POWER SUPPLY (+15V, 5W)	1 piece
3. RS232 CABLE	1 piece
4. CP76x software (CD)	1 piece
5. CP76x user manual	1 piece
6. Software (CD)	1 piece
7. CP76x user manual	1 piece

2. SOFTWARE INSTALLATION

This chapter describes the system requirements and how to install the software.

2.1 System Requirements

There are minimum hardware and software requirements that must be satisfied to ensure that the CP76x work properly. You must have:

- PC with Pentium processor, or higher;
- Windows-98, Windows NT 4.0, or higher;
- Monitor Resolution should be 800*600 or higher;
- Standard PC COM

2.2 Setup

Extract the WinZip file to a folder e.g. c:\CP76x (or other folder you specify). Run CP76X20.exe directly.

3. MAIN WINDOW

The main window is shown below:

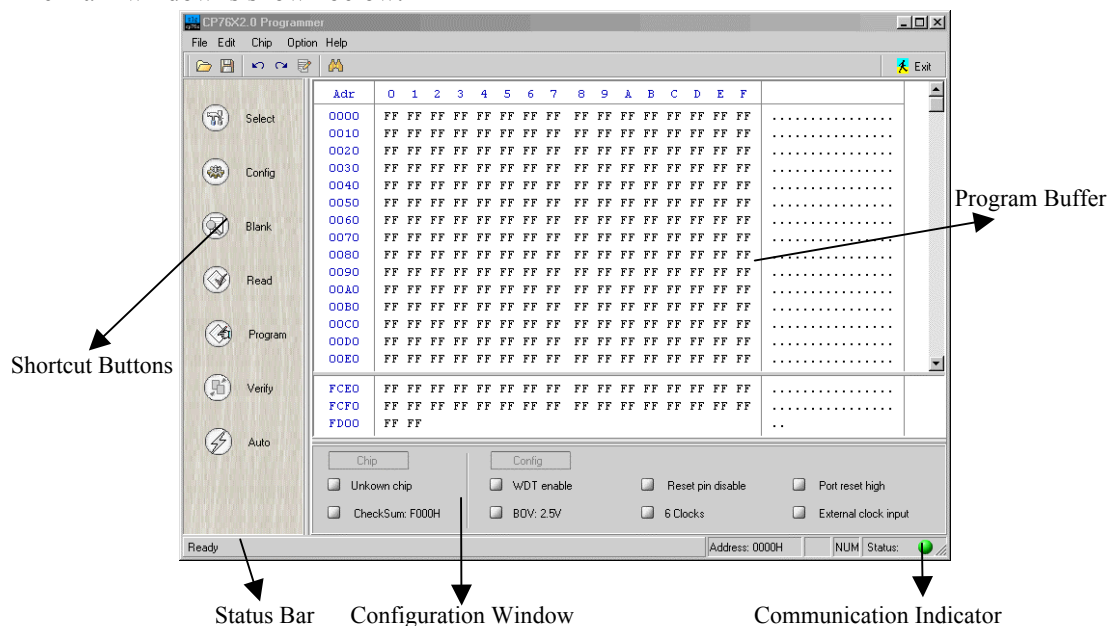


Figure 1. Main Window

Function Description

Shortcut Buttons

Frequently used functions. You can also use menu items to find relevant functions.

Program Buffer

Display program code.

Configuration Window

Display type of chip, checksum and other detailed configuration.

Status Bar

Display current operation, cursor position, status of Caps Lock and Num Lock key and communication indicator.

Communication Indicator

If system fails to connect with programmer, the color of indicator will be red. Otherwise it will be green.

4. Menu Functions

File

Open...

Load file into program.

Save as...

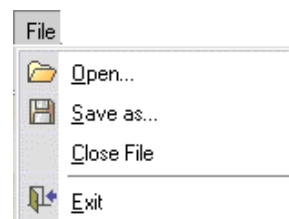
Save current file as Hex or Bin file.

Close File

Close current file.

Exit

Close CP76X software.



Edit

Undo

Ctrl + Z

Undo the modification of buffer.

Redo

Ctrl + Y

Redo the modification of buffer.

Find...

Ctrl + F

To find a Hex data or ASCII string.

Find Next

F3

Find next Hex data or ASCII string.

Edit Buffer

Enable you to edit program code buffer.

Fill with...

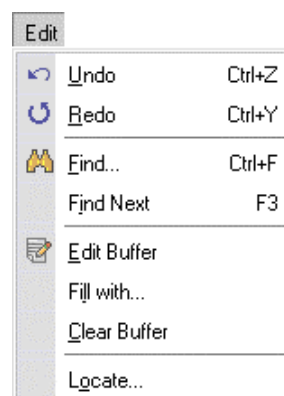
Fill specific segment with a fixed value.

Clear Buffer

Fill whole buffer with 0xFF.

Locate...

Locate cursor to specific address.



Device

Select...

Select which type of device you want to program.

Read...

F4

Read the content of device in adapter.

Program...

F5

Program the code in buffer into device in adapter.

Blank Check...

F6

Check if current device is blank.

Verify...

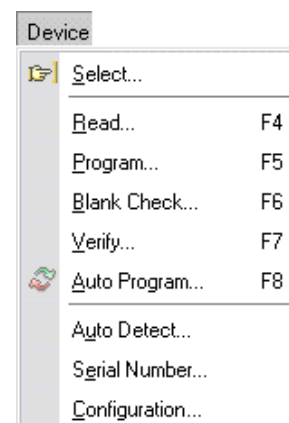
F7

Compare the content of current device with content of buffer including configuration.

Auto Program...

F8

Execute auto program operation.



Auto Detect...

Detect type of device in adapter automatically.

Serial Number...

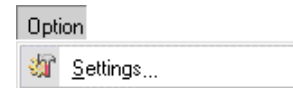
Set start address and initial value of serial number.

Configuration...

Set configuration word of current device.

Option**Settings...**

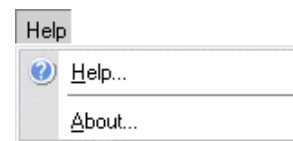
Configure display parameter of buffer, programmer parameter and other commonly used parameters.

**Help****Help...**

Display content of help file.

About...

About CP76X.



5. Function Description

5.1 Open file...

CP76x supports two types of data format. One is Intel Hex format. The other is Bin format. The following steps will show you how to load program code into buffer.

1. Select “Open” option from File menu, you will see below dialog:

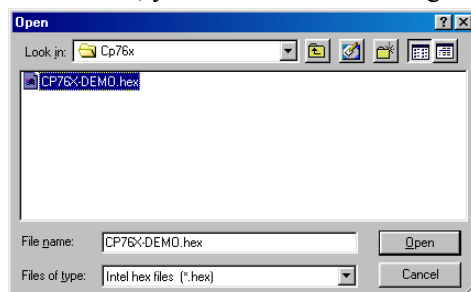


Figure 2. Open File dialog

2. Select the type of file. If you want to open binary file, please select “Binary files (*.bin)”.
3. Choose the desired file, then click "Open". The full filename and path will be displayed in the Title Bar of the program, the file will be loaded into memory and all displayed data in buffer will be updated.

Notes:

1. The length of data that is loaded into buffer should not exceed the limit of device that you select. For example, if you select P87LPC762 as current device, the program code that you load into program buffer should not exceed 2k, or the part of code that exceed 2k will be ignored.
2. If the program code (Hex format) contains data from address 0xFCE0h to 0xFD01h, CP76x will regard these data as user code and system configuration bytes.
3. If there is invalid data when you load file into program buffer, the rest of data will be ignored.

5.2 Save as...

You can save the content of buffer as Intel Hex format or Bin format. The following steps will show you how to load program code into buffer.

1. Select “Save as...” option from File menu,you will see below dialog:

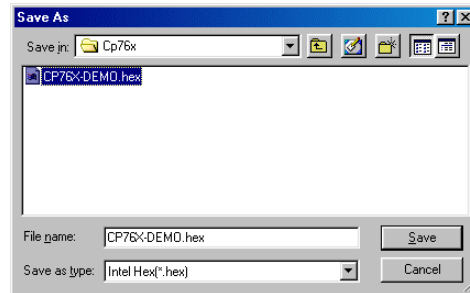


Figure 3 Save As dialog

2. Select the type of file. If you want to save the contents of buffer as binary file, please select “Binary files (*.bin)”.
3. Type filename and click “Save”.

Notes:

1. If you save the contents of buffer as a Hex file, the user code and system configuration word will be included.
2. If you save the contents of buffer as a Bin file, the user code and system configuration bytes will be lost.

5.3 Find

Following steps will show you how to find a HEX byte or ASCII string in program buffer.

1. Select “Find...” option from Edit menu, you will see below dialog:

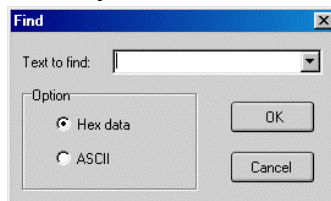


Figure 4. Find dialog

2. Select Hex data or ASCII. If you want to find a string, please select ASCII.
3. Input the data you want to find and click “Ok” button. Cursor will locate at address of the first matched data.
4. Click “Find Next” from Edit menu or F3 key to find next data.

Note:

When you select Hex data option, you can only input one byte of hex data. Non-hex character will be invalid.

5.4 Fill with...

This menu item is used to fill a specific segment of program buffer with a fixed value.

1. Select “Fill with...” option from Edit menu, you will see below dialog:

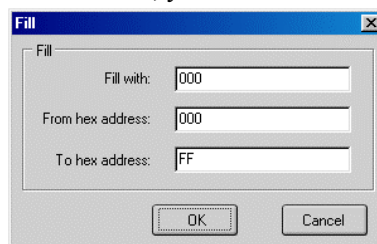


Figure 5. Fill With dialog

2. Input hex byte, start address and end address.
3. Click Ok button, data will be filled in the selected address and is shown in red color.

5.5 Locate...

This menu item is used to locate cursor to specific address in program buffer.

1. Select “Locate...” option from Edit menu, you will see below dialog:

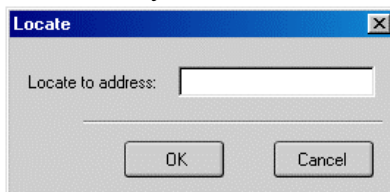


Figure 6. Locate dialog

2. Input desired address in dialog and then click “Ok” button or press “Enter” key.

5.6 Auto Detect

CP76x can detect the device in adapter automatically.

1. Select “Auto Detect” option from Chip menu, you will see below dialog:

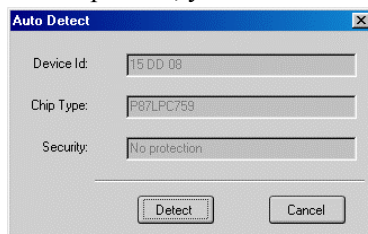


Figure 7. Auto Detect dialog

2. Click Detect button. If there is a 51LPC device in adapter, programmer will detect it and display the ID code, device type and security state.

5.7 Select

1. Click “Select” option from Device menu, you will see below dialog:

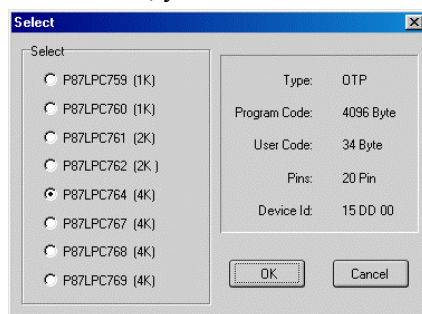


Figure 8. Select dialog

2. Select desired type of device in dialog, the basic information of selected device will be shown in right part of dialog.

3. Click Ok button or press Enter key.

Notes:

1. You should select proper type of device to be programmed. It's not recommended that use unmatched type to execute read or program operation.
2. After you select a device, the program buffer will be cleared. You should select device first, then load the desired program code into buffer.

5.8 Configuration Settings

1. Click “Configuration” option from Device menu, you will see below dialog:

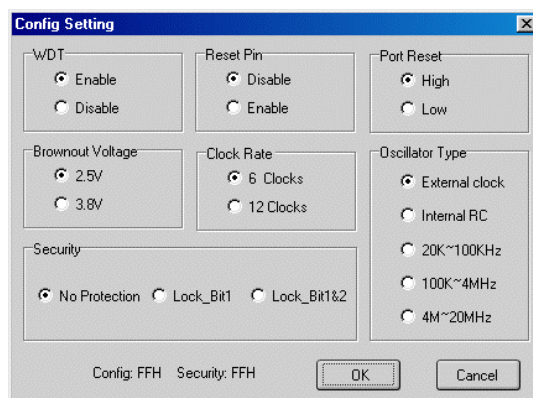


Figure 9. 51LPC Configuration dialog

2. Configure device according to your requirement.

3. Click “Ok” or press Enter key.

Notes:

1. Configuration word is very important to device. You should configure it carefully.

2. After you finished configuration, configuration window will show detailed information.

3. Security option:

- 1) No protection: Both security bits unprogrammed. No program security features enabled. EPROM is programmable and verifiable.
- 2) Lock bit1: Only security bit 1 programmed. Further EPROM programming is disabled. Security bit 2 may still be programmed.
- 3) Lock bit1&2: Both security bits programmed. All EPROM verification and programming are disabled (32-Byte Customer Code and system configuration bytes can still be readable).

5.9 Serial Number...

The serial number is used as an exclusive identifier of device. You can define the value and address of it.

1. Click “Serial Number...” option from Device menu, you will see below dialog:

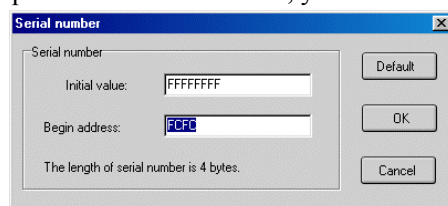


Figure 10. Serial Number dialog

2. Input initial value and start address in dialog.
3. Click Ok button or press Enter key.

Notes:

1. You should select device type before you enable this function.
2. Default value of serial number is “FFFFFFFF”. Default start address is “FCFC”. The length of serial number is 4 bytes.
3. Auto increment of serial number is available.
4. The lowest byte is located at the lowest address of 4 bytes. For example, serial number “12345678H” will be stored as: “ 78 56 34 12”.

5.10 Read

This item is used to read the memory contents of device, and writes them to the Buffer.

1. Click “Read...” option from Device menu, you will see below dialog:

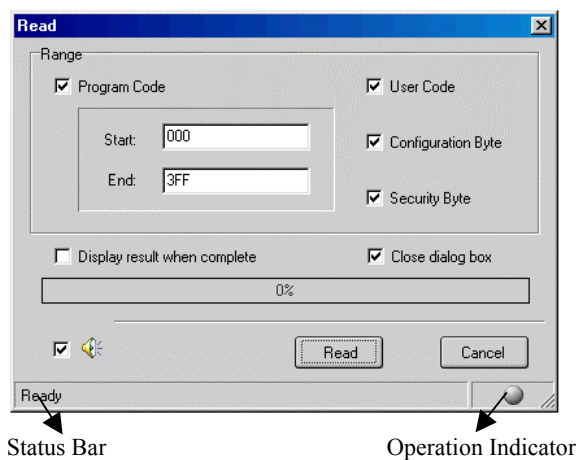


Figure 11. Read dialog

2. Set address range of read.
3. Select option(s) that you need.
4. Click Read button or press Enter key. The color of operation Indicator is red when read is in process. Indicator will blink if an error happens.
5. If you select option “Display result when complete”, a window will display the detailed information of read after operation is completed. You can click “Save” button to store it as a txt file.

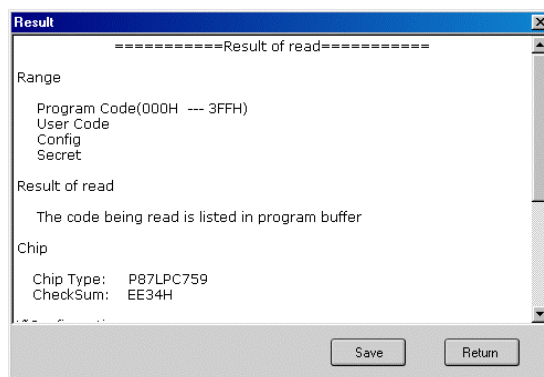


Figure 12. Result Of Read

5.11 Blank check

This item is used to check whether the current device is blank - that is all locations are set to 0FF Hex.

1. Click “Blank Check...” option from Device menu, you will see below dialog:

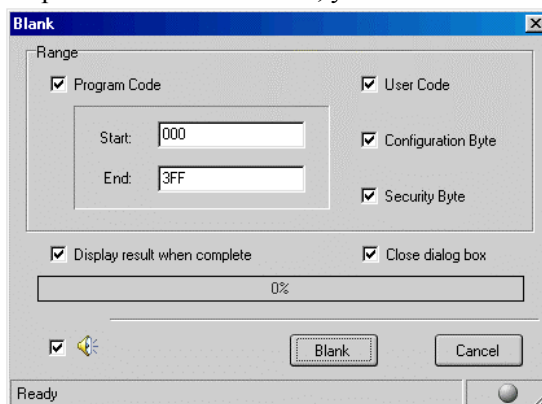


Figure 13. Blank Check dialog

2. Set address range of blank check.
3. Select option(s) that you need.
4. Click Blank button or press Enter key. The color of operation Indicator is red when blank check is in process. Indicator will blink if an error happens.
5. If you select option “Display result when complete”, a window will display the detailed information of check after operation is completed. You can click “Save” button to store it as a txt file.

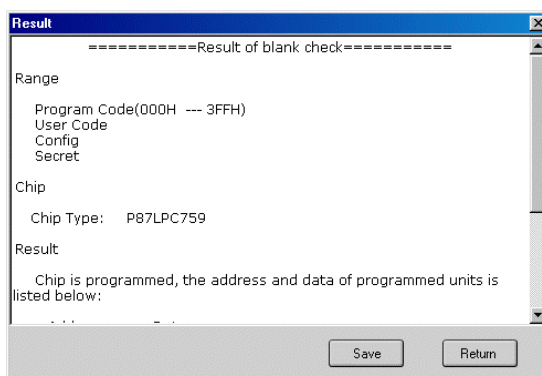


Figure 14. Result Of Blank

5.12 Verify

Compare the content of current device with content of buffer including configuration.

1. Click “Verify...” option from Device menu, you will see below dialog:

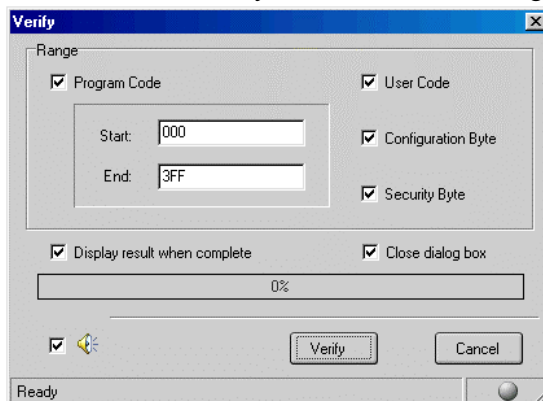


Figure 15. Verify dialog

2. Set address range of verification
3. Select option(s) that you need.
4. Click Verify button or press Enter key. The color of operation Indicator is red when verification is in process. Indicator will blink if an error happens.
5. If you select option “Display result when complete”, a window will display the detailed information of check after operation is completed. You can click “Save” button to store it as a txt file.

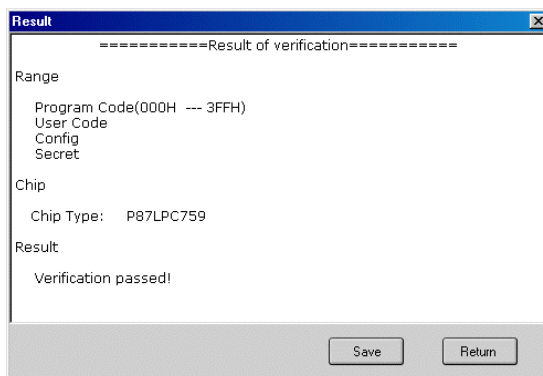


Figure 16. Verify dialog

5.13 Program

This item is used to program the current device with the contents of the Buffer.

1. Click “Program...” option from Device menu, you will see below dialog:

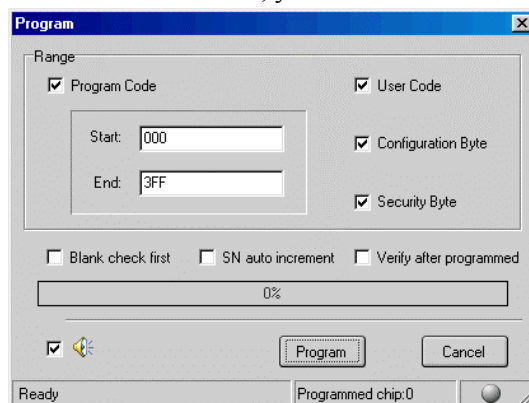


Figure 17. Program dialog

2. Set address range of programming.
3. Select option(s) that you need.
4. Click Program button or press Enter key. The color of operation Indicator is red when programming is in process. Indicator will blink if an error happens.
5. If you select “Blank before program” option and device is not blank, program operation will not be executed.
6. If you select “SN auto increment” option, serial number in buffer will increment automatically when the program of current device is completed.
7. If you select “Verify after programmed” option, verification will be executed when program is completed.

5.14 Auto Program

Auto program execute the same function as “Program” option. The only difference is that the program operation is executed automatically when you place device in adapter.

1. Click “Auto Program...” option from Device menu, you will see below dialog:

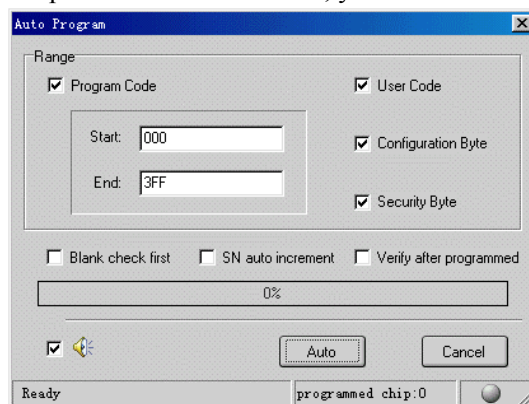


Figure 18. Auto Program dialog

2. Set address range of programming.
3. Select option(s) that you need.
4. Click Auto button or press Enter key.
5. System will detect if there is a new device in adapter after current device is programmed. When new device is detected, system will execute programming operation automatically. Note that the minimum time (in number of milliseconds) interval should be met. Refer to the section on Program Settings for details.
6. You can modify the initial count of chips being programmed. Refer to the section on Program Settings for details.

6. Settings

CP76X allow you modify some important parameters including buffer settings, program settings and other settings.

6.1 Buffer settings

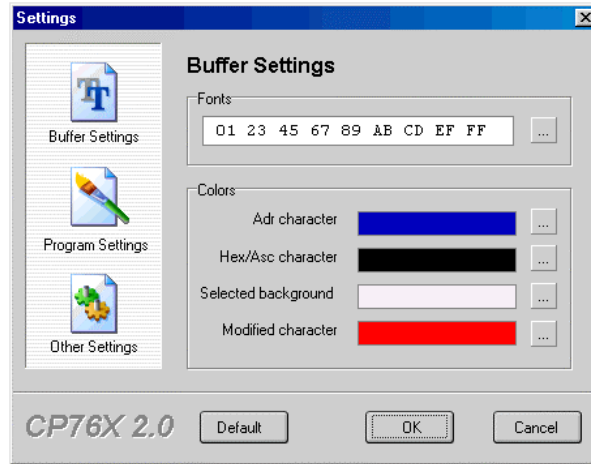


Figure 19. Buffer Settings dialog

Buffer settings allow you change the fonts and color of buffer.

6.2 Program Settings

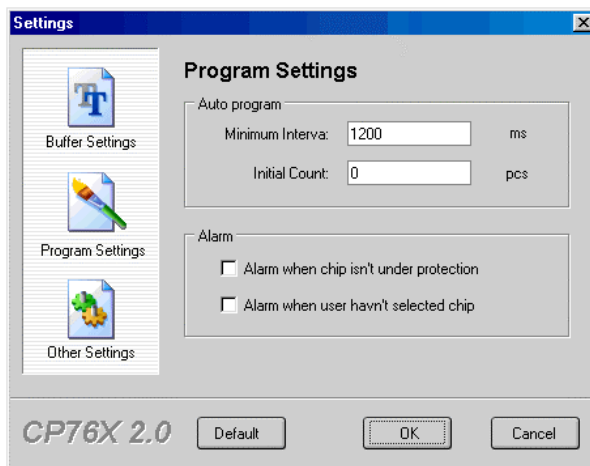


Figure 20. Program Settings dialog

Minimum Interval

1. Minimum interval is the time (in number of milliseconds) interval between the point that you remove previous device and the point that you install current device in adapter.
2. This option is valid only in auto program mode.
3. It's recommended using default value.

Initial Count

1. Initial count is the count of programmed chips when you start auto program operation.
2. This option is valid only in auto program mode.

Alarm

1. You can select whether system alarm when current chip is not under protection.
2. You can select whether system alarm when you have not selected type of chip.
3. These options are valid only in program & auto program mode.

6.3 Other Settings

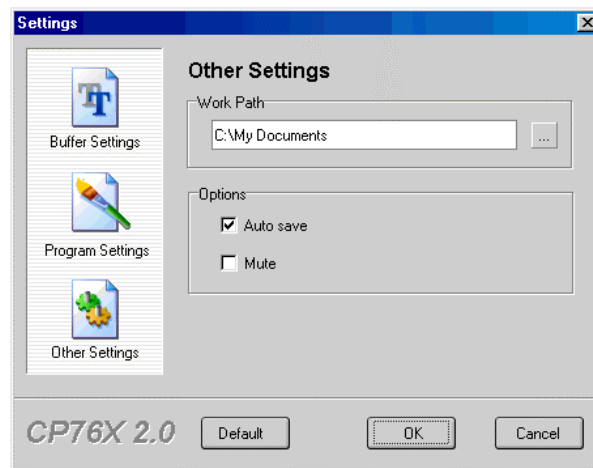


Figure 21. Other Settings dialog

Work path

1. The work path is changeable.
2. Default path is the directory of current software.

Options

1. If you select “Auto save” option, software will remember current settings when you open it next time.
2. If you select “Mute” option, all voice indication is inactive.

Appendix 1

Store configuration information in Hex file

51LPC series MCUs can only be programmed for one time. You should configure the configuration bytes correctly before you start to program chips. But if you need to configure the configuration bytes manually every time before you load program code into buffer, the probability of making mistake is rather high. It is recommended that you store the configuration information in program code, so that even an ordinary worker will be competent for programming work. In the following section we describe how to store the configuration bytes into your program code.

CP76x support two ways of storing the configuration bytes:

- If your program code has no configuration information i.e. there is no data at address 0xFCE0~0xFCFF and 0xFD00~0xFD01, you can load your file e.g. product1.hex into CP76X. Then open the Config dialog and configure the configuration bytes. After you confirm the correctness of these bytes, you can store the file in Hex format with another name e.g. product2.hex. Next you program device with CP76X, you need only load the product2.hex into program buffer. The configuration bytes will be loaded into buffer automatically.
- Add the configuration bytes into your source code. For example:

```

;*****
;
; 51LPC User Parameter & User Config Address Define
; User Parameter:    0FCE0~0xFCFFH    32Bytes
; User Config:      0FD00~0xFD01H    2 Bytes
;*****
USER_PARAMETER      equ    0FCE0H
USER_CONFIG1        equ    0FD00H
USER_CONFIG2        equ    0FD01H
;*****
;
; PHILIPS 51LPC Microcontroller USER CONFIG constant define
; UCFG1 define
; Address:  FD00H    Default value:  0FFH
; WDTE  RPD  PRHI  BOV  CLKR  FOSC2  FOSC1  FOSC0
;   B7   B6   B5   B4   B3   B2   B1   B0
;*****
WDTE_EN             equ    1000000B
WDTE_DIS            equ    0000000B

RPD_DIS             equ    0100000B
RPD_EN              equ    0000000B

PRHI_HIGH           equ    0010000B
PRHI_LOW            equ    0000000B

```

```
BOV_25V      equ    00010000B
BOV_38V      equ    00000000B
```

```
CLKR_PER6    equ    00001000B
CLKR_PER12   equ    00000000B
```

```
FOSC_EXTERN  equ    00000111B
FOSC_INRC    equ    00000011B
FOSC_LOW     equ    00000010B
FOSC_MIDDLE  equ    00000001B
FOSC_HIGH    equ    00000000B
```

```
,*****
;
; UCFG2  security define
; Address: FD01H   Default value: 0FFH
; SB2  SB1  —   —   —   —   —   —
; B7   B6  B5  B4  B3  B2  B1  B0
,*****
```

```
NO_SECURITY   equ    11000000B
ONE_SECURITY  equ    10000000B
TWO_SECURITY  equ    00000000B
```

```
org    USER_PARAMETER
DB     "ZLGMCU XXX_PRO VER 1.00 20020101"
```

```
org    USER_CONFIG1
DB     WDTE_EN or RPD_DIS or PRHI_LOW or BOV_25V or CLKR_PER6 or FOSC_INRC
```

```
org    USER_CONFIG2
DB     ONE_SECURITY
```

```
;
END
```

When you emulate your source code with these added code, some emulation software may reports error of program space overflow. You can use semicolon to mask these code. When you finish your debug, you can recover them and create the Hex file.

Appendix 2

An example of programming 51LPC series MCUs with CP76x

We take example for CP76X_Demo.hex to describe the step of programming:

1. Open CP76X software.
2. Select device via shortcut button or menu item.
3. Load CP76X_Demo.hex file into program buffer.
4. Configure the system configuration bytes.
5. Store current content of buffer with name CP76X_Demo1.hex.
6. Confirm the correctness of check sum.
7. Start programming operation.
8. Test the first programmed device in actual circuit.
9. Test one programmed device every a few (e.g. 200) pieces of device.

Notes:

1. Next time you can load CP76X_Demo1.hex and ignore the step 4 and 5.
2. If the quantity of devices to be programmed is large, it recommended that use auto program to improve the efficiency.