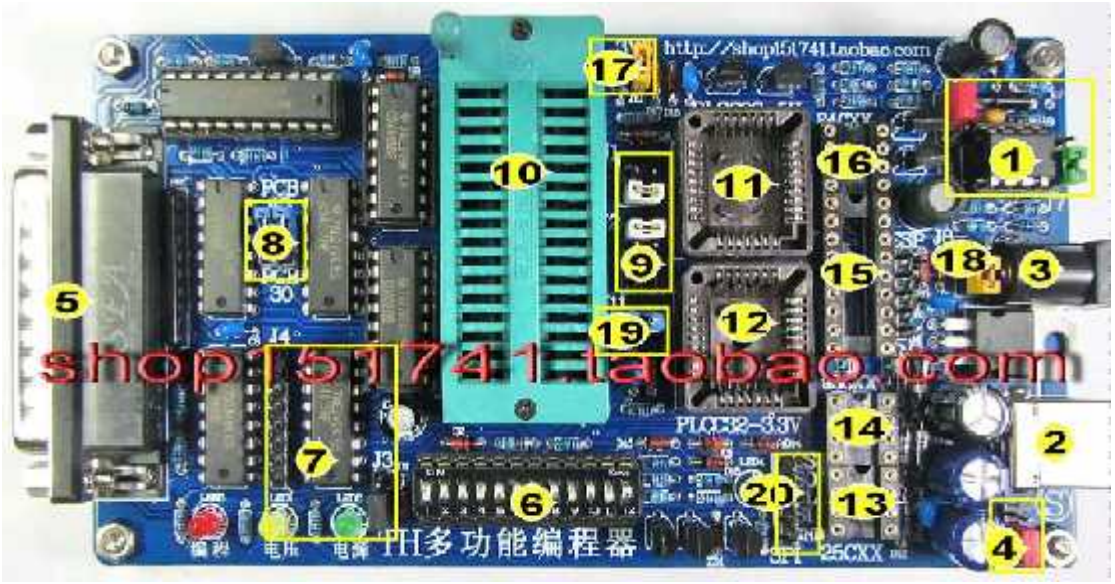
TH Multifunction Programmer

TH Multifunction Programmer (PCB5.0C): Hardware Installation and Setup

First: TH Multifunction Programmer:

Introduction of the architecture of the hardware.



1. **Programming voltage jumpers (J6 & J7)** to set the **programming voltage** of the **27 series chips.**

   Note that this is the **default jumper setting**.
   Output voltage is **12.5V.**

2. **USB power supply interface:** Use the supplied USB connection cable to connect the motherboard USB serial interface to a PC. TH *provides the operating voltage*.

3. **9-15V power supply interface: Use an external power supply** for the programmer. Can be **AC** or **DC,** as appropriate.

4. **Power supply selection jumper (J12).** Set the jumper to select either **USB** or **external *power supply mode.***

   Settings Jumper 12

Bottom Position: **USB supply (5V) from PC**

Top Position: **External Power supply** (**12V, - Pin in Centre**) via **Jack 3**

Attention: No Simultaneous connection of USB and an external 12V power supply .

5. **25-pins parallel connector:** connect the programmer via a parallel cable to a random parallel data communication port of a computer.

6. **DIP switches (12)**: Used i**n 27 series, 28 series, 29 series, 39 \ 49 series IC's**, such as memory chips.

   Before reading and writing operations, you need to dial these switches to the appropriate positions for the chip.

   For the specific settings Refer to the display of the software.

   Specific jumper Software or manual methods are detailed in the display in the Annex.

7. **Chip expansion jumpers (J4 & J3)** for **4M or 8M chips**, and high-capacity data line expansion.

8. **Hardware version conversion jumpers (J5 & J9):** hardware version **PCB 5.0** programmer circuit design, to **PCB 3.0.**

   When hardware **conversion jumper**, is set to **PCB30 position**, you can use the **previous 0.97JA** and **other versions of older drivers**.

9. **Programming jumpers (J1 & J2)** for setting several special chips: **2732**, **2716**, **2816**, **I28F001**, **AT29C256**

10. **32 PIN zero insertion force IC socket (ZIF32):** for **27 series**, **28 series**, **29 series**,

    39 \ 49 series, such as memory chips and the **PLCC chips adapter suitable for ATMEL 8, 9 series** microcontrollers

    And Set Up of the **FWH / LPC series chip adapter**.

11. **PLCC chip socket** for programming **28, 29, 39, 49 series** voltage of **5V PLCC32** chips.

12. **FHW / LPC chip socket** for programming **FHW / LPC** series **3.3V PLCC32 chips**.

13. **Socket for 25 series memory, SPI Flash** memory series (**25CXX**).

14. **Socket for 93 series** memory (**PIC93CXX**).

15 **Socket** for **PIC MCU chips (16FXX)**, a full range of PIC microcontrollers adapter.

16. **Socke**t for **24 Series** memory (**24CXX**).

17. **Jumper** (**J10**) for **3.6V, SPI Flash memory** series.

18 **Jumper** (**J8**) for **6.5V, VCC chip.**

19. **Jumper** (**J11**) for on-board **FWH / LPC adapter** switch (Remove an external FWH / LPC adapter).

20. **Connecting pins** for the **interface board**.

21. **LED 1 (Yellow):** Programming **Voltage Low**: Use external Power Adapter

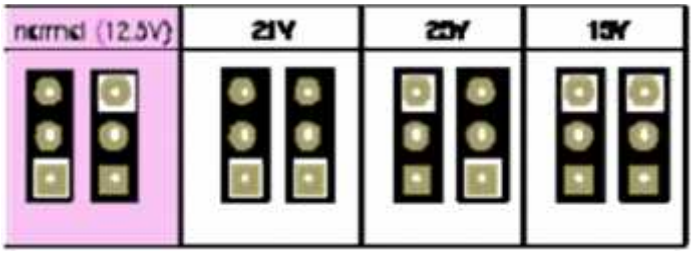
LED 2 (Green): **Power On**

LED 3 (Red): **Wait**: Programming Ongoing

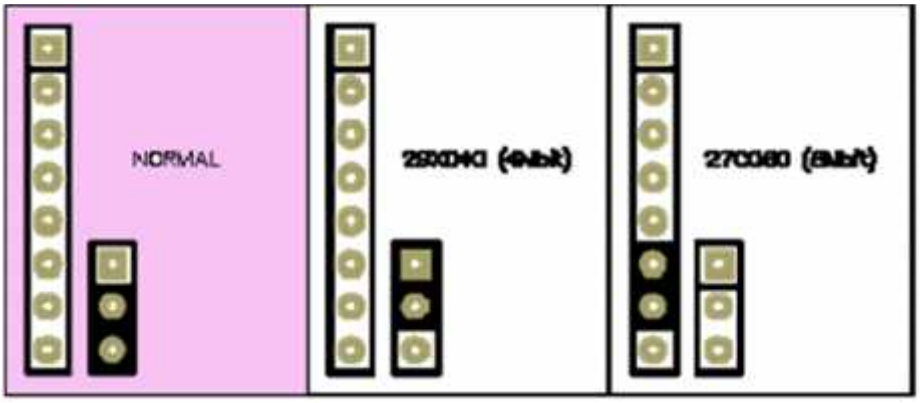
<u>Second</u>: Setting of the programming jumpers

Note: The following settings are *advanced settings*, **ordinary users do not need to set the following jumpers**

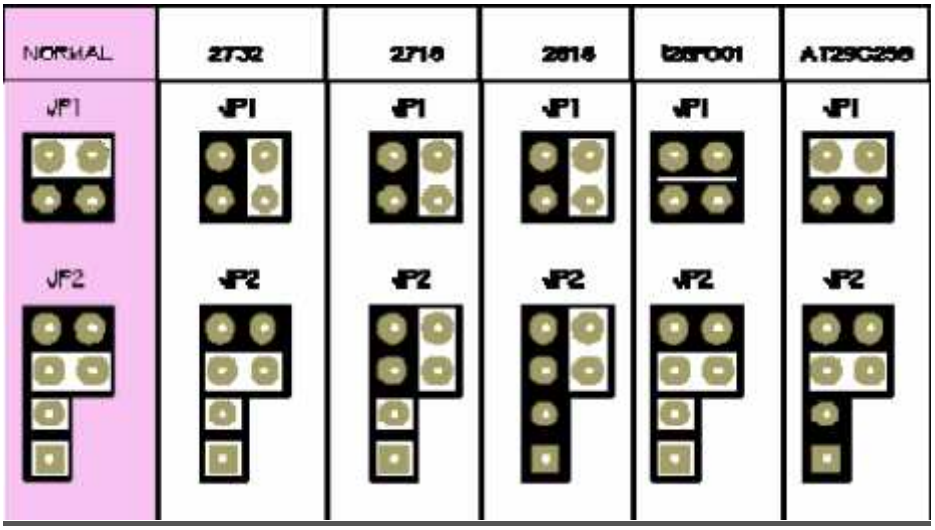
[Programming voltage jumpers J6 & J7]



[Chip capacity expansion jumpers J4 & J3]



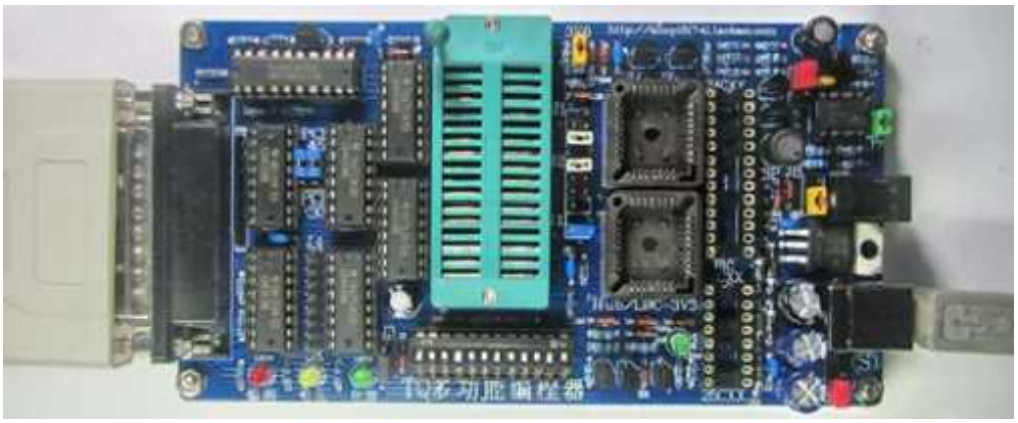
[Chip programming jumpers J1 & J2]



Third: Installation of the TH Programmer hardware

Installation steps of the TH Multifunction Programmer are as follows:

1. Turn off the computer
2. Plug a parallel cable into the computer's printer port
3. Plug the other end of the parallel cable into the 25-pin connector of the TH programmer



4. Install Programmer supply

   A. Use an external power supply: (**12V DC**)

   Plug the power adapter into a 220V power outlet; plug in the power output into **Socket 3 (Negative on Central Pin).**

   In this case, the position of the **jumper 12,** should be placed in the **upper position**;

   B. Use USB power: Plug the USB cable into a random USB port of the computer; Insert the other end of the USB cable into the **USB port** of the TH Programmer (**Socket 2**);

   The position of **the jumper 12,** should be placed in the **lower position**.

Note that the External supply and the USB power supply, cannot be used simultaneously!

This will damage the Programmer!!

Next the programmer's **power indicator** (**LED 2, green**) should light up, indicating that the **programmer is powered on**.

The Hardware connection is now complete.

Note: Set the computer's parallel port to **ECP** or (**ECP + EPP)** in the **BIOS of the PC**.

Fourth: Installation of the control software

Run the appropriate version of the **setup.exe** **installation software program** located on the supplied CD-ROM (latest version **0.98D11**).

Follow the instructions step by step operation to complete the driver installation.

After the installation is complete, double click the shortcut link Icon on the desktop

Eprom50.lnk

to run the **Eprom50** program.

Driver default installation directory is **C: \ Program Files \ EPROM50**

Fifth: Detection of the TH Programmer Hardware

After starting the control program, *just select a chip model*, (**you must select a chip model in the program)**

Help item in the menu bar, click "*detection hardware items*",

if the power supply is *not properly connected*,

a "*hardware error*"

or

"*data communication error*" message is displayed,

you have then to carefully check whether the programmer and the host is connected,

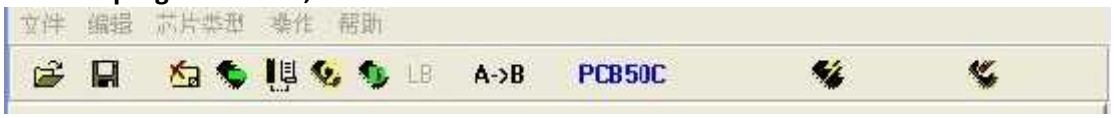
and the power supply is normal (**12V DC external** or **USB** (5V), depending on **Jumper 12**).

If the Programmer and power supply is normal, the "*hardware detected*" prompt appears.

TH driver screen and Programmer Features:

When the **EPROM50 Control software** is running, the following screen appears:

Toolbar: program screen,



the *second line* is the **toolbar**,

the meaning of each **icon** is as follows:

Reads a file, the data is read *into the program's cache*. For example, you have earlier downloaded a BIOS Upgrade file. It could be e.g. a binary (BIN ...) or a HEX file (HEX ...).

Supported types: Intel HEX (* .hex); Binary (* .bin); Motorola S Record (* .s);

ALL Eprom Files (* .bin, * .hex, * .s)

Save a file, such as the programming contents of the cache memory *into a bin file.*

Clear memory. Data programmer *cache emptied*.

Read operation. Reading of a data chip into the programmer's cache.

Verification operation. Comparing whether the data in the cache and the data written into the chip are the same.

Blank check operation. Test whether a chip has been fully erased.

Display chip signature.

Read and display the 28 Series, 29 Series, 39/49 series, such as memory chips, 89 MCU, AVR MCU chip electronic signature.

Programming / testing. The contents of the buffer into the chip. For SRAM test function to perform.

Erase operation. Erase chip content.

Display DIP switch status, we have been in a **PCB3** status below to illustrate each DIP switch should be in position.

Programming Configuration bits. Programming **MCS51 series**, **AVR series** single chip configuration bits.

Menu bar:

In the *program screen*, in the *first column* of the menu bar, the meaning of each item is as follows:

[File]: open, save a file and exit the program.

[Edit]: Edit data in the cache.

[Device]: Chip Type: *Select the chip.* It shows a list of supported chip programming, where you select the chip type to be programmed. You can also use shortcut keys to select the menu bar.

[Action]: You can select the following operations: Read; Programming / test; verification; blank check; erase;

Read the configuration bits; programmed configuration bits; display chip signature;

[Help]: Detection whether programmer hardware *connection* is correct;

Detection of *programmer hardware*.

Detecting the various states programming: testing supply, programming voltage, and that various signals are normal;

Copyright information

Main picture area: the main screen of the program, is divided into four regions, from left to right, top to bottom,

As follows:

1. *chip select,* the chip area and the schematic of the DIP jumper settings;

2. the jumper schematic for the required chip and for the appropriate Adapter;

3. the setting area of the chip parameters;

4. the setting area of the programming parameters;

1. *the chip selection and parameter area*: Here, you can choose the type chip to be selected and programmed.

   After selection of the chip model, the parameters of the corresponding chip will appear; it will also show the corresponding DIP switch settings for that chip.

2. *a schematic diagram of the required chip adapter or jumpers*: you have to set the jumpers according to the schematic displayed here.
   The various jumpers are corresponding with their location on the drive.

3. *the chip parameter setting area*: Special parameters corresponding to a specific chip, will appear here; Encryption and decryption of some of the setting parameters.

4. the *programming parameter* setting area: some fine-tuning of the programming operation;
   Generally these do not need adjustment (working in the background).

In the following, a window is shown with the series of selection buttons.

Press the "**buffer zone**" button to **display the contents of the buffer**.

The first column of the data address, the last one for the data in **ASCII, hexadecimal** value will be displayed in the middle of the data address.

For a PIC microcontroller having an internal EEPROM, the content of EEPROM data area will be automatically displayed.

Status Bar:

The last line of the screen, the status bar is used to **display the status of the program**, such as **Core Sheet is not written correctly**, **the Listing / position of the chip programming errors, any programmer failures**.

Chip installation in the TH Multifunction (PCB5.0C) Programmer

Since there are several BIOS chip packages, and different types of chips, the number of pins and pin definitions are quite different.

Thus, to use the corresponding adapter socket, the TH Multifunction Programmer has integrated sockets specific for the chip; therefore for a particular chip, select the socket based on the type corresponding to that socket.

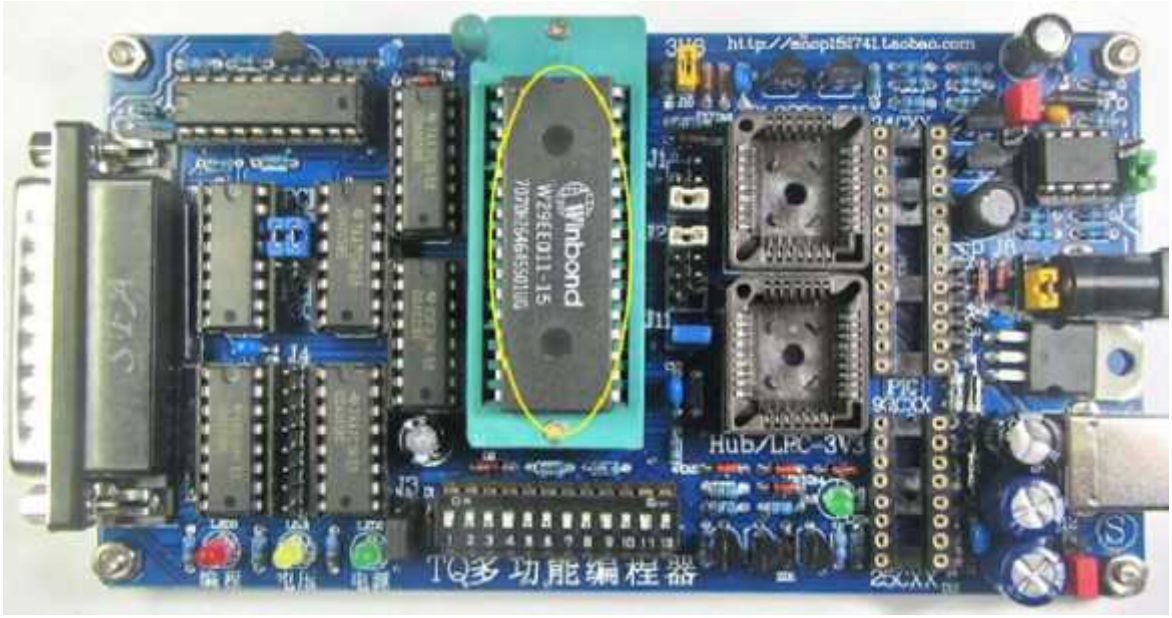
At present for current BIOS chips, there are different mounting positions on the programmer for respectively **sub-5V, 3.3V FWH / LPC chips** and **27 series EPROM chips;**

Please refer to the following descriptions:

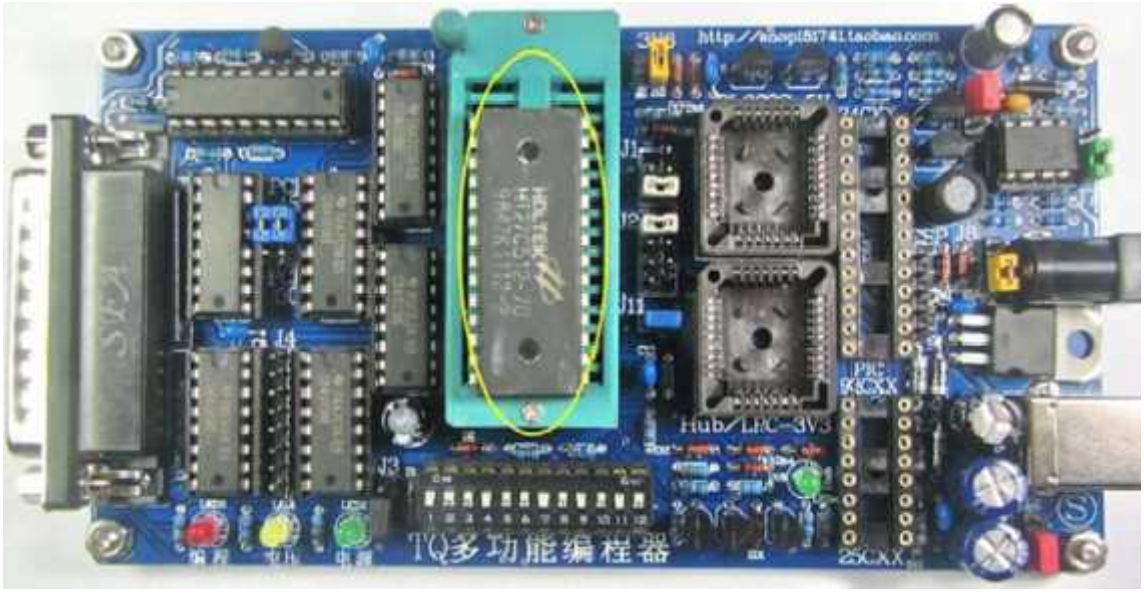
3.3V FWH / LPC chip (**PLCC32-3.3V**)**:** mounting position on the programmer

5V PLCC chip (PLCC32-5V): mounting position on the programmer

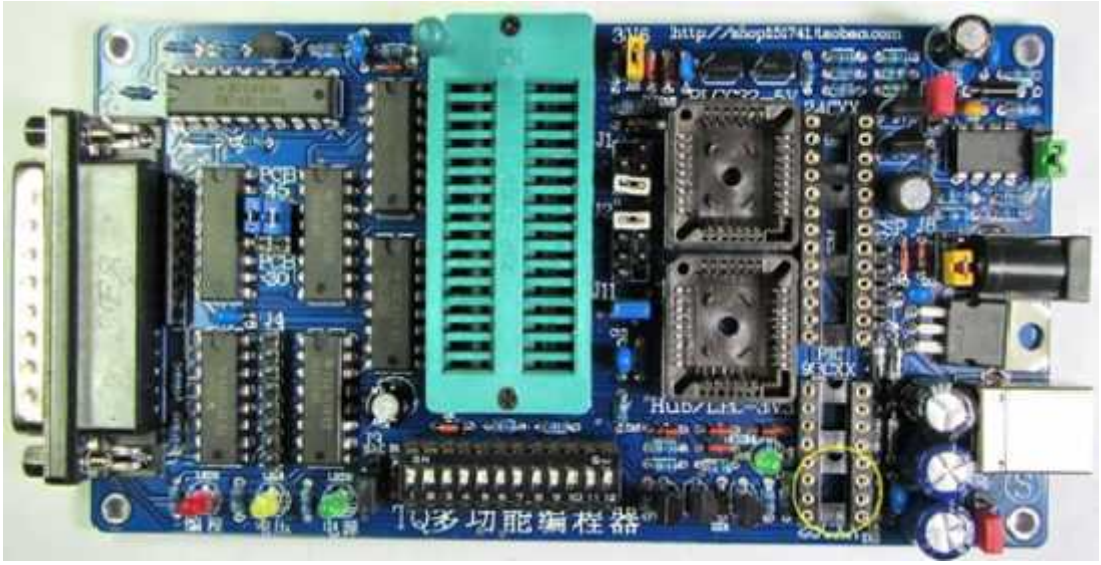
5V DIP chip: mounting position on the programmer:



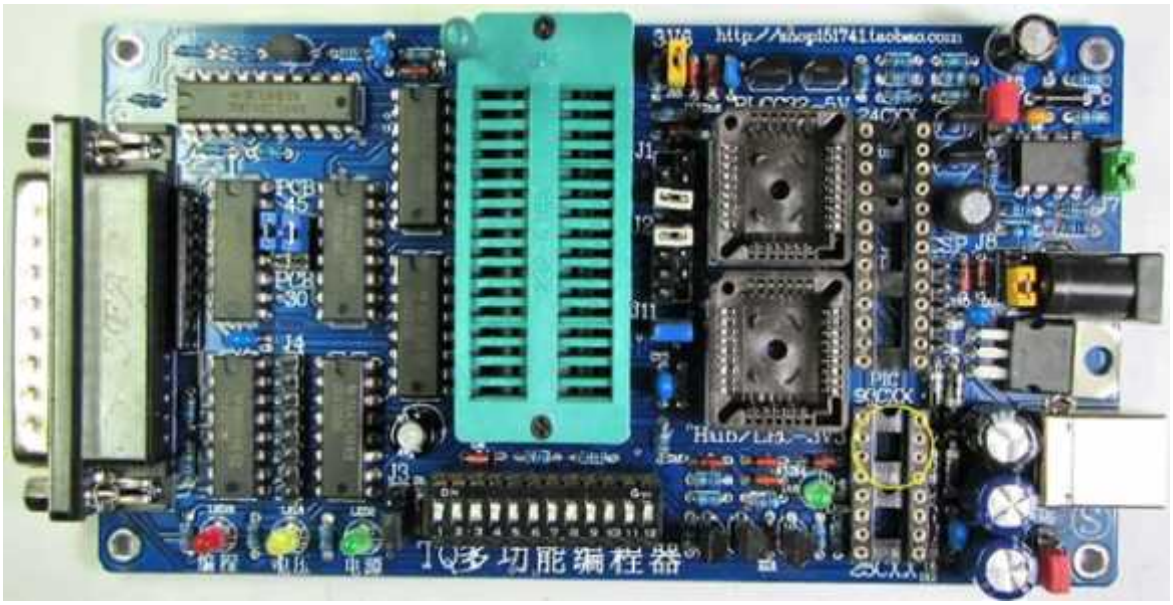
27 series EPROM 28-pins chip: mounting position on the programmer:



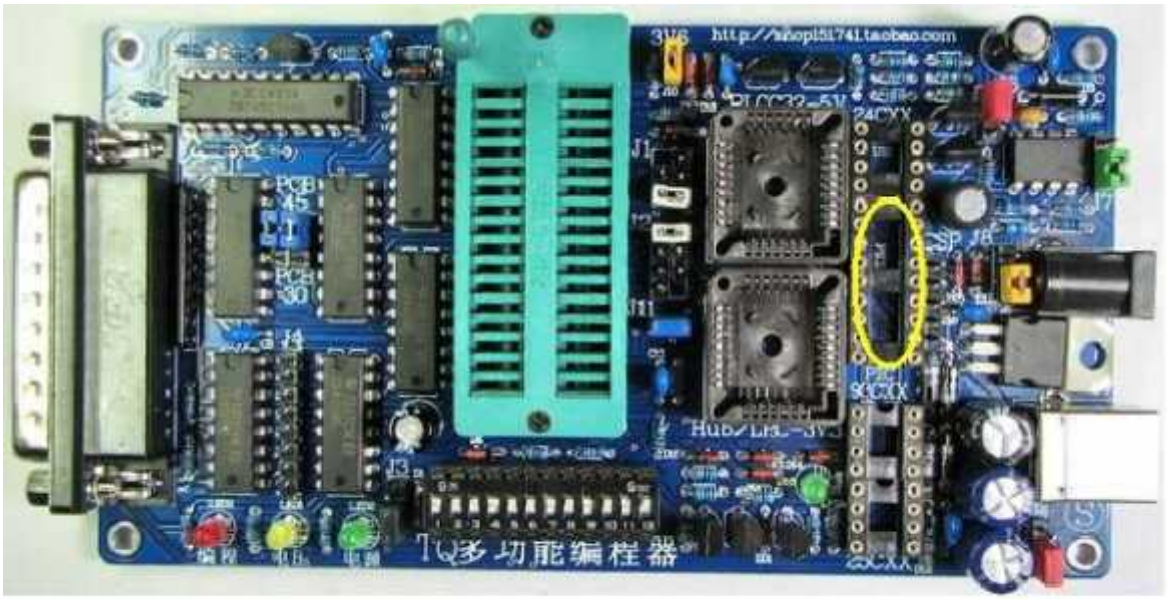
25CXX, SPI Flash series chip: mounting position on the programmer



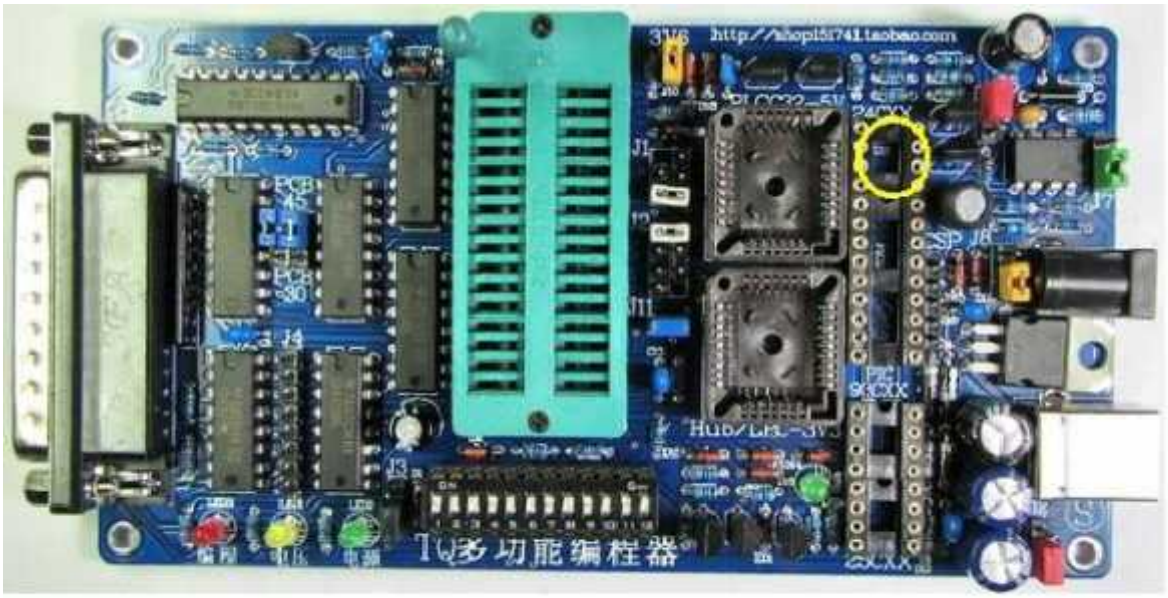
93CXX series chip: mounting position on the programmer:



PIC MCU (**16FXX**) chip: mounting position on the programmer:



24CXX series chip: mounting position on the programmer:



Setting and programming of the TH multifunction BIOS (PCB5.0C) Programmer

The programming (“writing”) of the TH multifunctional programmer for BIOS chip programming is very simple.

In particular, as long as we manually select the correct chip type, set up the relevant jumpers after having been prompted by the program and the required chip is plugged in the correct location (socket), and having acquired the BIOS file to be transferred, you can easily read and write operations onto the chip.

Here we give as an example the specific operation to write a **SST49LF002A** (3.3V voltage) chip for the Inte**l845** motherboard:

Note:

for the programming of a **SST49LF002A** chip, you **need to employ the default programming voltage** of **12.5V;**

Therefore, in order to prevent **high breakdown voltage** of the chip, each jumper on the programmer must be **in the default location!**

The **SST49LF002A** to **FWH 3.3V PLCC chip**, must first be plugged into the TH multifunction programmer **FWH** socket (**PLCC32-3.3V**), as shown below:

Start the software Program “**EPROM50”**, click on **Chip Select**, select **Firmware ->**

HUB / LPC -> **INTEL / ATMEL** -> **Intel SST49LF002A**.

Press the **chip select button** (**Device**) option, also present on the **Chip Select** menu bar.

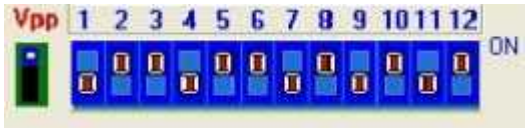
In the next figure, the setting position of the various DIP switches is shown:

you have to Click on the **icon of the DIP switches** on the programming set.

The Jumper (**J3**) switch next to the schematic diagram also appears. It has to be set accordingly.

The 12 white dip switches of the Programmer, must correspond with the position of the switches in the software.

=



DIP switch setting is as follows: DIP switch to **TOP**: the switch is **ON**, DIP switch to the **BOTTOM** is **OFF**.

In the above diagram, the status of the DIP switches is:

OFF, ON, ON, OFF, ON, ON, OFF, ON, OFF, ON, OFF, ON.

The **Chip select buttons** below shows the **parameters of the chip,**

(generally speaking, there is no need to adjust the parameters of the chip; you can use the system default values.)

Size / check value (Size & checksum):

displays the capacity and checksum value of the chip and data buffer.

Address Properties (Shift & pattern address):

displays address range and the highest ground Address bits of the Chip to be programmed .

Write pulse properties (tWP / WC):

displays the write pulse width and write latency of the chip to be programmed.

After setting the programmer, we can **back up the chip** or **refresh the program.**

1. **The backup file**

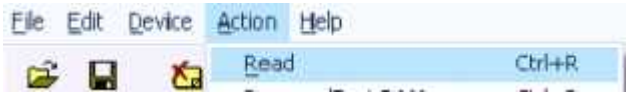
If you have downloaded the content of the chip, we **can read the contents of the chip**, and **save it as a file**.

You can **also write these directly to the other chip.**

Read the file

Select Drive menu bar operations [Action] - **Read**,

Or you can click "Read toolbar

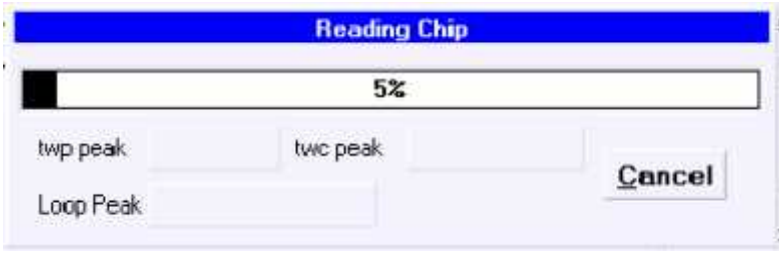


Or Button “read-out chip content. “

Read out the contents of the Chip and stores these into the buffer. If the buffer contains several (earlier) data, it will overwrite those data.

You can click on the program screen **"buffer"** item below, and drag the scroll bar to view all the data in the buffer.

In the process of reading the chip program, the Red LED (LED 3) , next to the Yellow LED (LED 1) is illuminated.



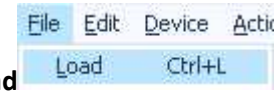
Save document

To save the contents of the buffer (as it has been read out from the chip) ***to a file,*** select the **[File] – Save As**

Toolbar or select "Save" Button. ***Thus, you can save the file.***

2. Chip Programming

Choose a correct type of chip for the chip to be programmed and after having inserted this chip correctly, click on the menu bar

[File]--Load 

or

"Open File" button on the toolbar, and you write

The file name

The file is now read into the buffer.

(The above file can e.g. be downloaded from the Internet where you can obtain upgrade files from the Manufacturer, or it can be *the file read from the chip and which you earlier saved as a backup file*).

Note: the **FWH / LPC chip** must be **empty** before being written, so we must ***first erase*** before ***write***-

As follows:

Click **"Erase"** on the toolbar or the Button **to erase the contents of the chip**, and then use a "**blank check**" or the button **to detect whether the chip is empty**.

Then, **"Programming"**

Click on the Button, **the data is written from the buffer chip**.

After the programming has been completed, the program will automatically check the written data.

In the process of programming the chip, **the red light (LED 3) is on**;

if the chip also requires **Vpp programming voltage, the yellow indicator (LED 1) will light.**

This process takes few system resources; in the process of programming the chip, you can open another window, and perform other operations.

3. Copy the Chip

First select the chip type, then plug in the original (“source”) chip. Using the **"Read"** function, the program reads the original chip’s content into the Buffer.

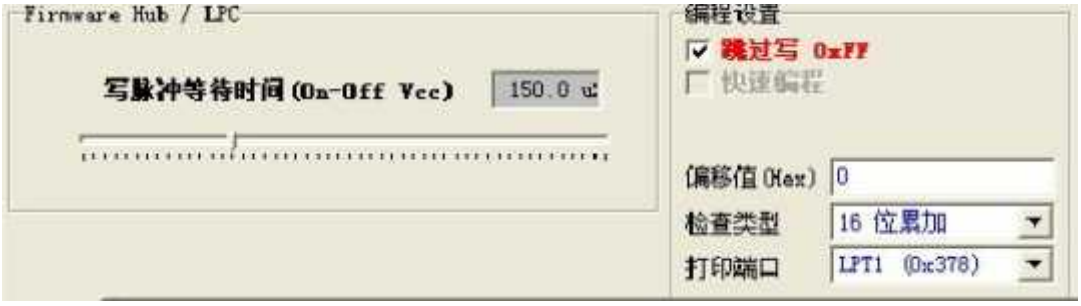
Next put in the empty chip (“target”) to write it, using the "programming" function to write data.

Note: If you choose the wrong chip model, or insert it in the wrong position you are likely to burn (destroy) the chips.

The following on-chip programming parameter settings are for advanced users who can use the following Operation:

If the chip is **programmed correctly**, please **do not make** the following adjustments;

For the **novice user**, **do not adjust** the following parameters.



RC delay time (write pulse waiting time): Adjust programming pulse delay.

If your computer is fast or the programmer is not working properly, you can adjust a large delay value.

Skip Write 0xFF (skip FF): Choose whether to **skip** the programming content for a **FF** byte.

Fast Programming: choice is **not** fast programming.

Printer Port (Port Select): Select **Parallel multifunction programmer** connection.

Typically **LPT1** (**0X378**).

Offset (Offset): Set the address where the programming begins.

Check Type (verify mode): You can select **32-bit** or **16-bit** CRC accumulation mode.

Programming of **SST** or **SST** Series **FWH / LPC** chips: you can choose the following options to speed up the read and write speed.

Note:

1. Some chips are only written correctly after the first erasing sequence, so preferably erase the chip empty before the first chip programming operation.
2. When the user has selected the chip, and then the program prompts, properly set up the 12 dip switches.
3. In the process of starting the machine, the working condition of the programmer is unstable. Therefore, wait for the machine to enter the WINDOWS interface, start the software program EPROM50, select the correct model of the chip.

   Next, the chip can be plugged into the programmer.
4. Programming the chip to read and write operations, only ads operating voltage to the chip;
   - If the **Yellow** LED on the programmer lights, do not remove the chip.
   - If you want to *interrupt the operation*, press the *End button*;
   - After the reading and writing the chip, and on-chip voltage (Yellow) has disappeared, the new chip can be inserted and then written.

     This is **not a hot-swappable.**

Programming Operation of **EPROM27 series chips** (**E**lectrically **E**rasable)

Some EPROM chips, such as **W27C512** or **W27C512**, (albeit being in the **27 series**) do ***not*** have an ordinary EPROM chip erase window (for UV erasing).

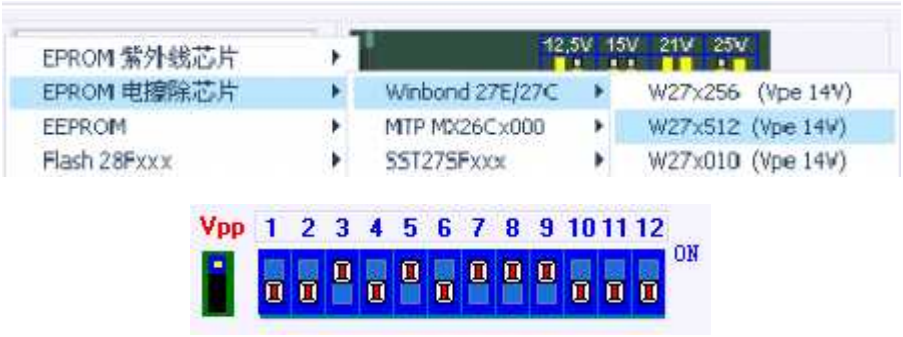
Such chips, are called **Electrically Erasable EPROM** chips. The programming operation of such chips is troublesome (having to adjust the voltage jumpers), and incurs a certain degree of risk!

As an example, a detailed description of the procedure of a **WinBond El**ectrically **E**rasable chip **W27C512** chip, is presented as follows:

1. **Select chip and settings**: **WinBond chip Electrically Erasable**

   The TH Multifunction Programmer supports chip models: (W27E512, W27E010, W27C010, W27C020, W27C040)

First, start the program **EPROM50** , press the **Select button** -> EPROM chip electrically erasable chips -> Winbond 27E / 27C - > W27X512



The setting of the DIP switches is as follows:
DIP switch to the TOP switch is **ON**, to the BOTTOM is **OFF.**

Hence, in the above diagram, the settings of the DIP switches are:

OFF, OFF, ON, OFF, ON, OFF, ON, ON, ON, OFF, OFF, OFF.

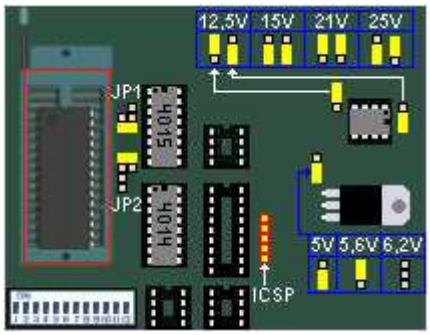
The setting of **Jumper 3 (J3) ->** (**Vpp**) is also shown.

2. **Determine the chip location**:

The program shows the location of the chip on the **32-pin ZIF socket**.

The program also shows the setting of **Jumper 1** (**J1**)**, Jumper 2 (J2), (Default setting)**

Make adjustments accordingly.



Note: The program stipulates the chip position, in the programming ZIF socket.

3. The chip erase operation:

Chip correctly plugged into the ZIF socket of the programmer, and **J1** and **J2** jumpers set correctly;

Before writing to the **27 chip**, you **need to empty the chip**;

click the program tool **Erase button bar**

,

the system will display the following erase setting interface:

Select Settings

Before selecting Erase Chip, an erase interface appears wherein the **2/3 PIN** and **DIP SW setting** boxes have **not** been selected.

The **Chip Erase button is greyed out and cannot** be **selected**; this is because the system determines whether the setup of the jumpers has been correctly performed;

Because the **chip 512** is **electrically erased** when erasing, the **JP1** needs to be revised, namely:

An extra jumper jumper ("Erase Unprotect") *has to be added* to **short pins 2-3**,

(this is the **new red jumper**):

Install the new **Red JP1 jumper** on the programmer **to short pins 2-3**, set the **White JP1 and JP2 jumpers** (NB **White** on board - **Yellow** in the program) correctly (default).

Tick the box 2/3 Pin and *box* **DIP SW setting in the software program window.**

Click on the "**Chip Erase**" **button** (not grayed out, any more),

The chip can now be erased correctly.

4. Chip programming operation:

After **Chip Erase** and **Empty detection** is completed, the program will set the interface to appear as follows

"Erase Chip" should be "**Remove jumper**" !

Select Settings

"Erase Chip" should be **"Remove jumper"** !

<u>Before programming</u>, there is the **need to remove** the extra **RED JP1 jumper** **("Erase Unprotect")**;

This drawing, is presented as a reminder ***to remove the added* JP1 jumper (between Pin 2/3).**

De-Select **2/3 PIN**, click [**remove jumper**] button, **to *return back to the editing screen***:

Select **File** –> **Load** -> **Select**, or click on the icon on the toolbar;

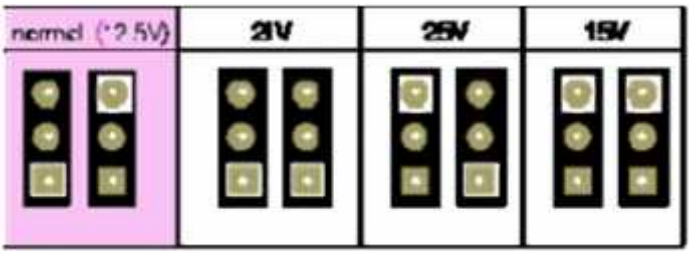
The BIOS file (to be written) is transferred to the Cache.

Then click

on the toolbar, the cached BIOS File is *written into the Chip*.

The *Correctness of the Write* is subsequently *Verified*:

Note:

Different types of chips, require different **Erase Voltages**.

Depending on the chip, the erase voltage needs to be set **by two jumpers J6 & J7**, to obtain the required operating voltage (**default is 12.5V**).