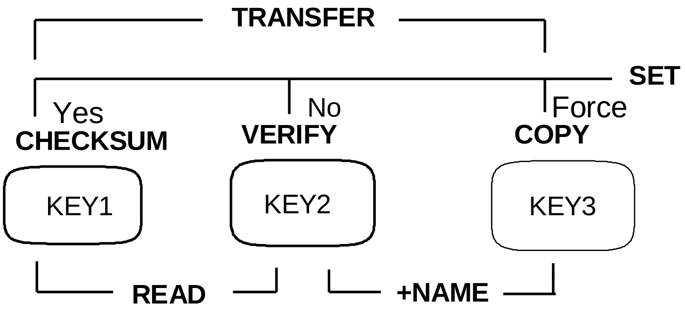
# *UNIV2000/2000A/2000B Operation*

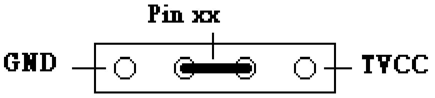
## Key layout:



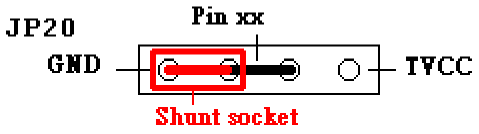
## Pin-jumper setting:

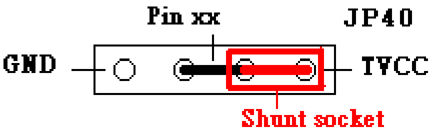
Because the pin-jumper setting is a very important step when using this programmer, we make some detailed explanation here:

Along the side of each textool socket, there are many sets of square pin terminal labeled **JPxx**. Each set is composed of 4 horizontally aligned pins. For every set, **the leftmost** pin is the **GND** and **the rightmost** pin is the **VCC**, **the middle two** pins(they are shorted internally) is wired to one **pin** of the textool socket.

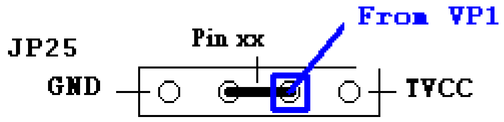


Take an example, if the LCD shows a message of "G:J20"(which means the pin at the JP20 has to be set to GND), a jumper has to be used to short the JP20's leftmost two pins so that the textool's pin which was wired to the JP20(doesn't has to be the pin20 of the textool) was connected to GND. Similarly for message "V:JP40", a jumper has to short the JP40's rightmost two pins to make one of the textool's pin connected to VCC.





For the other three sets of jumper located at the upper side of the textool, labeled **VP1, VP2** and **VP3**, are for the VPP control. All 4 pins of each set are shorted internally and was wired from a programming high voltage source. A message of "VP1:J25" means the pin at the JP25 has to be wired to the supply source VP1. A wire with both of its end hooked with a jumper was used to connect between the **middle two pins** of the JP25 terminal and any two pins of the **VP1** terminal.



PDF created with pdfFactory Pro trial version www.pdffactory.com

For example, if the LCD display shows

```
W29C020  DIP32?
V:J46   G:J26
```

The first row message means "do you want to handle a DIP32 packaged W29C020"? The second row message means "Set the terminal **JP46 to VCC** and the terminal **JP26 to GND**". To comply it, we have to use a jumper to short the rightmost two pins of the JP46 terminal, and another jumper to short the leftmost two pins of the JP26 terminal. After doing this, depress the key-switch **key2**(the middle one) to check if there are any further settings was required. If all has been done, depress the key-switch "Yes"(**key1**, the left one) to confirm this setting. If all the jumpers was set correctly, the programmer will accept this device for further processing. It will also display the **checksum value** of the internal buffer and its **ID**.

If the setting is incorrect, the programmer will make two beeps to warn the user. The user has to reset those jumpers and try again.

Sometimes it might be needed to ignore the current settings(such as for operations which are jumper-setting irrelevant) and to enter the operation mode directly. At this situation, the user can force the programmer to accept whatever the current setting is by depressing the key1 and key3 together.

PDF created with pdfFactory Pro trial version www.pdffactory.com

## Operation commands:

### 0. TRANSFER(depress key1 and key3 together)

After TRANSFER command was entered, there are three function items(CHANGE GROUP, DOWNLOAD and UPLOAD) to choose. They are circulated displayed by depressing the key1 and then confirmed by depressing the key3. This command can be abort by depressing the key1 and key3 together.

0.1 **CHANGE GROUP**

When this function was shown on the display, depress key2 can change the group number.

0.2 **DOWNLOAD**

This function is for receiving data downloaded from the PC's COM port and store it into the internal buffer. The data length is decided by the PC's file size.

0.3 **UPLOAD**

This function is for transmitting data to the PC's COM port. The data length is decided by the current device size.

### 1. +NAME(depress the key2 and key3 together)

Select the device name for processing.

After enter this mode, each depress of key2 and key3 will bring out the next name(in order of device size and alphabetic order of the name). Each depress of key1 and key2 will bring out the previous name. Use key2 to check all the required pin-jumper settings. Use key1 to confirm the settings.

### 2. CHECKSUM(depress key1)

Calculate the buffer/device checksum value.

If the textool socket has **no device** inserted **or** the device **has pin-contact problem**, this calculation will be conducted on the **internal buffer**(the master)with message ***"Buffersum...*** ". Otherwise, the **device**'s checksum was calculated with message ***"Slavesum...*** ".

During the time of calculating, the device's ID CODE will be temporally shown on the display(at the master's checksum location) until the checksum value was calculated completely.

### 3. BLANK CHECK(depress and hold key1 until a beep sound)

To check if the device is blank or not. For some devices with protect feature, it also check its protect status.

### 4. VERIFY(depress key2)

To verify if the content of the internal buffer is the same as the device. For some devices with protect feature, it also verify its protect status respect to the internal PSB value.(Hint: This command was best suitable for checking the device pin contact).

### 5. COPY(depress key3)

This command has a series of actions. First it check the device's protect status and clean it. Second it checks if the device is blank and erase it if required.(For some devices, it might doing the erase without the blank check.) Third it copy the content of the internal buffer to the device. Fourth it verify the content. At the last it might doing the protect action according to the internal PSB value.

PDF created with pdfFactory Pro trial version www.pdffactory.com

### 6. ERASE/FORCE(depress and hold key3 until a beep sound)

[Erase]: For devices which can be electrically erased, this command can erase it and clear the protect status.

[Force]: For devices which can not be erased electrically, this command will bypass the blank check action and doing the copy directly.

### 7. READ(depress key1 and key2 together)

Read the content of the device and store into the internal buffer.

When this command was executed, the programmer will ask the user to do a reconfirm to avoid accidentally destroy the content of the internal buffer.

For some devices, its protect status will also be read and stored as the internal PSB value.

**Note:**

1. The way the programmer stores data in the internal buffer is different between the 8bit device and the 16bit device. You can't READ a 16bit devices and COPY the content to a 8bit device, or vise-verse.

One possible solution is that: After the READing, upload the content to the PC first and then change to the new device name, download the data and then do the COPY.

### 8. SET PSB(depress all three keys together)

For devices which has write-protect or security feature, this command is for its setting. At the final step of the "COPY" command, programmer will set the device's status according to this internal PSB values.

Depressing all three keys together will bring out the PSB value in bit form on the LCD display. It has totally 48 bits, shows in six 8bits sets. For Flash memory, each bit corresponds to one can-be-protected sector, the rightmost bit is of the one which is at the lowest address and the leftmost bit is of the highest address one. For other type of ICs, each bit is for their corresponding fuse(refer to the file DeviceNote.txt for detail). A bit 0 means the fuse should be set to an un-lock or default status after the COPY command. A bit 1 means the fuse has to be locked or activated at that time. Use [key1] or [key2] to move the cursor to the proper bit position and use [key3] to toggle it status. When the cursor moves out the on-display bits region, it will bring out the next 8bit set for further processing. After all bits are set, move the cursor to any empty position and depress [key3] to end the setting procedure. The new value will be shown on the display immediately.

This value will be stored nonvolatile until it was modified by the user or by a READ command. It was shown in two HEX digit on the left corner of the LCD display. Usually a 00 value means that this IC should be set to a non-lock or default status when the COPY command was conducted. When this position is left empty, it means there is no related features build-in in this IC or, the programmer didn support it yet.

PDF created with pdfFactory Pro trial version www.pdffactory.com

# Download/Upload data

## *A: RS232 port(for UNIV2000/2000A)*

### *Download data From Programmer to PC:*

1. Connect the RS232 cable between the PC COM port and the UNIV2000 programmer. Run the univ2000.exe program, choose the option->port to set the correct **COM** port.

2. At the programmer side, depress **key1+key3** to enter the transfer mode. Using key1 to bring up the download item, using key3 to execute. After that, the LCD will shows "DOWN LOAD RDY" and waiting for data transfer.

3. At the PC side, choose the file->download to set the required filename and then tick the start button. After complete, the programmer's LCD display will shows "Download Complete!".

4. The size stored in the programmer is the same as the file be downloaded, the way the data stored is decided by the name the programmer current selected. You can't download data under a 8bit device name and then change to a 16bit device name to do the copy, or vise-verse.

### *Upload data from the programmer to PC:*

1. At the programmer side, depress **key1+key3** to enter the transfer mode. Using **key1** to bring up the upload item, using **key3** to execute. After that, the LCD will shows "UP LOAD RDY" and waiting for data transfer.

2. Connect the RS232 cable between the PC COM port and the UNIV2000 programmer. Run the univ2000.exe program, choose the option->port to set the correct **COM** port.

3. Choose the file->download to set the required filename and then tick the start button. After complete, the LCD display of the programmer will shows "Upload Complete!!!".

4. The stored file size is the same as the IC size the programmer currently selected.

## *B: USB port(for UNIV2000B)*

1. Connect the USB cable between programmer and PC. The first time user need to complete the driver's installation first(see install.doc).
2. At the programmer side, depress **key1+key3** to enter the transfer mode. Using **key1** to bring up the "UpLoad Rdy" or “DownLoad Rdy” message, using **key3** to execute.
3. At the PC side, execute DPL245.EXE to bring out the operational window
4. Depress the “**Connect DLP245**” button, make sure the “Connection OK” message shows on the status bar
5. Enter the required Upload or Download **filename**.
6. Depress the “**Upload**” or “**Download**” button to start the transfer.
7. After finished, depress the “**Close**” button to end.

**Note: you must close the DLP245 window first before disconnect the USB cable.**

PDF created with pdfFactory Pro trial version www.pdffactory.com

# UNIV2000A/2000B Firmware Update

## *Using the AVR2(LPT port only) or AVR3(also the USB port support)*

Using the firmware update cable(5pins plug -- 25pins D type) to connect the PC printer port and the UNIV2000A/2000B. Turn on its power. Run the AVR2.exe(or AVR3.exe) on the PC.

1. Select the required module file(binary type) through the **Open** button.
2. Select the **Port** address and ATMega128 **Type**, un-check all the **FUSE** and **LOCK** box.
3. Press the **PROG** button to start the update
4. After complete, the **FUSE** and **LOCK** status will be read-back and shown on the right side the value box. If there is no problem, the machine will be re-boot and running. Press the Exit button to end.

### Trouble shooting on using the AVR2:

If something is wrong and the machine is not re-boot, check the read-back FUSE and LOCK value with the correct one:

| | |
|---|---|
| Fuse Lo | “E0” |
| Fuse Hi | “D1” |
| Fuse Ext | “FF” |
| Lock | “FF” |

If they are different, check the required box and fill-in the correct value than re-PROG again.

**Note:**

1. You need purchase the USB2LPT dongle to use the USB port firmware update.
2. AVR2/AVR3 can also be used to update the firmware of UNIV2000, but the CPU Type has to be “ATMega103”, the Fuse should be set to “0xDF” and the module file has to be converted into .bin format before usage.

PDF created with pdfFactory Pro trial version www.pdffactory.com

# Appendix A: **Programming Example(MX29LV160B)**

## ***1. settings***

Choose the name MX29LV160B through **+NAME** or **-NAME** (sometimes you might need to change to the proper group first). The bottom line of the LCD display will show the required jumper setting

**V:35  G:38,53**

Use jumper to connect **JP35** to **VCC** (short the rightmost two pins).
Use jumper to connect **JP38** and **JP53** to **GND** (short the leftmost two pins).

Using **VERIFY(NO)** key to bring out further settings if there is any. The LCD will show

**VP1:J7*  VP2:J32***

Use jumper wire to connect **VP1**(any two pins of four) and **JP7**(the middle two pins).
Use jumper wire to connect **VP2**(any two pins of four) and **JP32**(the middle two pins).
Depress **CHECKSUM(YES)** key to confirm the settings.
*refer the **pin-jumper settings** section in the operation manual*

If the setting is correct, the programmer will have a short beep, display the device ID and recalculate the internal buffer's checksum value. If not, there is a two beeps warning and waiting for further jumper set/correction.

## ***2. Writing***

***case 1: data is in a master IC***

Clip the Master IC into the TSOP48 socket, depress **CHECKSUM** key. The LCD will show **"Slavesum...."** and start the calculation of the IC Checksum. During the calculation, the IC's ID will be displayed on the upper right corner. After finished, the calculated checksum will be displayed on the lower right corner and the internal buffer checksum will be on the upper right corner.

If you know the calculated checksum value is correct(sometime you might not), depress the **READ** key. The LCD will show "Confirm Read ?" to make sure you are going to overwrite the internal buffer content. Depress **Yes** if you are (or **No** to abort). After finished, the new buffer's checksum value will be recalculated and displayed for reference. You can use the VERIFY to make sure the content is correct.

Replace the Master IC with another one and depress the COPY key to start the programming.

If there is contact problem between IC and socket, the LCD will show **"Buffersum...."** and the internal buffer checksum will be calculated. During the calculation, the "should-be" IC's ID will be displayed on the upper right corner for reference.

(Tip: leave the socket empty when depress the checksum key will calculate the internal buffer content)

(Tip: you can use **VERIFY** key to quickly find out if there is any pin in bad contact condition. If any pin goes wrong, you will know its pin number immediately)

***Case 2: Data is in the disk***

You need to download the required data to refresh the internal buffer. Please refer to download section in this operation manual.

PDF created with pdfFactory Pro trial version www.pdffactory.com