

GQ USB Universal Programmer

User Guide



Revision 4.05

Nov 2018

Revision History and Acknowledgement

- Re 4.05 Content revised for installation. By MCUmall Nov 2018
- Re 4.04 Revised for GQ-4x4(GQ-4X Version 4) by MCUmall Aug 2015
- Re 4.01 Revised USB driver installation note by MCUmall June 2012
- Re 4.00 Revised for new software by MCUmall March 2009
- Re 3.01 Revised by Robert W. Meister September 2008
- Re 3.00 Revised for GQ-4X by MCUmall August 2008
- Re 2.00 Revised for GQ-3X by MCUmall July 2007
- Re 1.00 Initial version for GQ-2X by MCUmall May 2006

Table of Contents

1. INTRODUCTION	5
2. USB DRIVER AND SOFTWARE INSTALLATION	5
2.1 USB DRIVER 3.0 OR LATER VERSION INSTALLATION	6
2.1 APPLICATION SOFTWARE INSTALLATION	8
2.2 START THE PROGRAMMER'S SOFTWARE	8
3. HARDWARE	10
4. SOFTWARE INTERFACE	10
4.1 GENUINE PROGRAMMER VERIFICATION	11
4.1 BASIC MENU AND THE TOOL BAR ITEM	13
4.1.1  LOAD / SAVE DATA FILE	13
4.1.2 <i>Modify the data buffer</i>	13
4.1.3 <i>Set the data buffer starting offset address in the buffer window</i>	13
4.1.4 <i>"Edit" menu item</i>	14
4.1.5 <i>Fill Buffer</i>	14
4.1.6 <i>"Find"</i>	14
4.1.7 <i>The pop-up menu</i>	14
4.2  DEVICE TYPE SELECTION	15
4.2.1 <i>The My Favorite Devices List</i>	15
4.2.2 <i>Searching a devices from the device list</i>	15
4.3 TEST H/W MENU ITEM	16
4.4 SOFTWARE STATUS BAR	16
4.4.1 <i>Checksum</i>	17
4.4.2 <i>Programmer status</i>	17
4.4.3 <i>Programmer revision</i>	17
4.5 DATA BUFFER TABS	17
4.6 DIP SETTING INDICATION (GQ-2X ONLY)	17
4.7 OPERATION SPEED MEASUREMENT	17
4.8 THE ADAPTERS	17
4.9 THE AVR ISP PROGRAMMING	18
4.10 THE JTAG PROGRAMMING	18
5. DEVICE OPERATION	20
5.1 STEPS	20
5.2 FILE OFFSET	20
5.3 DEVICE OFFSET	20
5.4  WRITE (PROGRAM)	20
5.5  READ	21
5.6  ERASE	21

5.7		BLANK	21
5.8		VERIFY.....	21
5.9		LOCK/PROTECT	22
5.10		READ ID	22
5.11		SPEED	22
5.12		AUTO MODE	22
5.13		DEVICE LOCATION IN ZIF	23
6.		THE DEVICES.TXT FILE	23

1. Introduction

The new generation GQ USB universal programmer is designed to be a portable and convenient device. The programmer uses the USB port for both power and communications, so it will work on a laptop that only has a USB port. It integrates a high-speed MCU with USB connectivity on board. The programmer GQ-4x4(GQ-4X Version 4) is hardware compatible with the Willem EPROM Programmer, so it will accept existing Willem EPROM Programmer's adapters (except the FWH/HUB adapter which needs a simple modification to increase the speed).

The programmer uses an on-board MCU timer to operate the target chip, so it will be much more stable than a parallel EPROM programmer. On-board dual power management lets the user control the voltage independently during programming, verification, blank checking and reading operations. The programmer also has an external DC input jack for situations where an un-powered USB hub is used.

The programmer software also provides many new features, such as:

- Auto-detection of the programmer model and hardware revision.
- Binary data editing capabilities, such as Find, Search, Copy and Paste.
- Device search feature to provide a fast way of finding the device model.
- The Favorite Devices feature provides another quick way to let you store frequently used target devices, thus eliminating redundant searching each time.
- The software provides short cut key to all operations.
- A Auto Run feature for quick batch operations.
- The user is able to add new device IDs to the ID library and device list.

The GQ USB universal programmer software supports Windows XP / VISTA /Win 7/ Win 8/ Win8.1/Win 10 (both 32-bit and 64-bit).

IMPORTANT

In order to le programmer work, both USB driver and application software need to be installed.

The USB driver have to be installed manually and it can be downloaded from:

http://www.mcumall.com/comersus/store/mcumall_download.asp

2. USB driver and Software Installation

The software can be installed before you connect the programmer to a PC. The USB driver installation needs the programmer must be connected.

2.1 USB driver 3.0 or later version installation

The USB driver 3.0 or later version is a Microsoft digitally signed USB driver. It works on Windows XP SP2, Vista t, Win7, Win8, Win8.1, Win 10 Windows systems for both 32 and 64 bit Windows.

When you first time connect the programmer to Windows, the Windows should detect a new hardware and launch the driver installation wizard.

If you missed USB driver installation wizard, then you can install the USB driver from Windows device manager.

See following Reply #6 for the details.

http://www.mcumall.com/forum/topic.asp?TOPIC_ID=2030

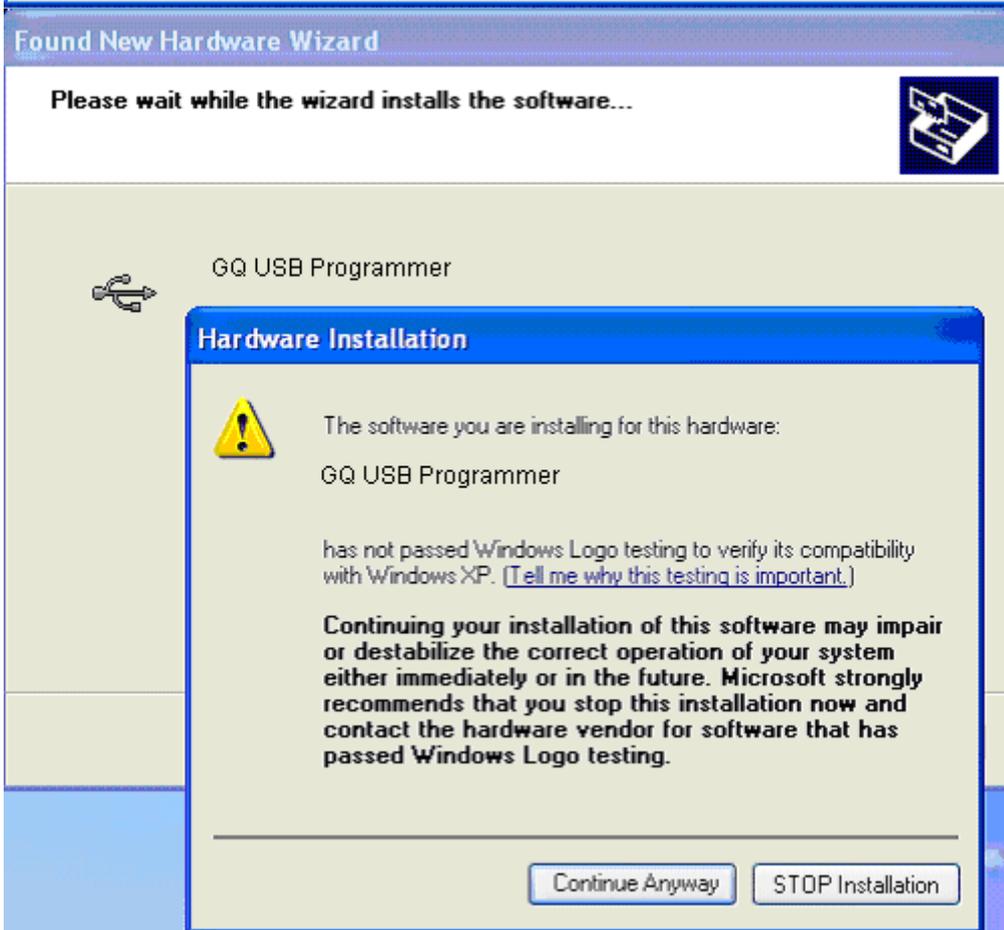
There is a video shows how to do it. See:

<https://www.youtube.com/watch?v=w1Iow7KWhsA>

The driver needs to be installed manually, either from Windows New Hardware Installation Wizard or Update the driver from the Windows Device Manager.

For Windows XP you will see these installation screens:





Click on "Continue Anyway" to finish the driver installation.



- Restart the computer if the computer prompts you.
- The programmer's green power LED should light up.

2.1 Application software installation

1. Download software GQUSBPrG and run USBPrGSetup.exe. Follow the installation instructions. USBPrGSetup.exe will install both the application and a USB driver for the programmer.
2. Once the installation has completed, connect the GQ USB programmer to the PC's USB port.

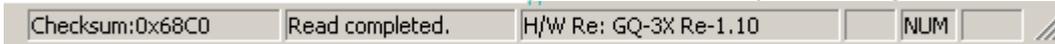
Note: If you are using an un-powered USB hub or extension hub, it may not provide enough power to operate the USB programmer. In this case, you may need a 9V, 200mA (or greater) DC adapter with a center-positive, 2.1mm DC plug.

2.2 Start the programmer's software

1. Run the GQ USB Programmer Software from the Windows menu: Start → Programs.



Wait for 5-10 seconds for the "Programmer: Ready" message to show up on the status bar.



2. You are ready to program the device.

3. Hardware

There are three LEDs located at the left bottom of programmer. Those LEDs indicate the current state of programmer.

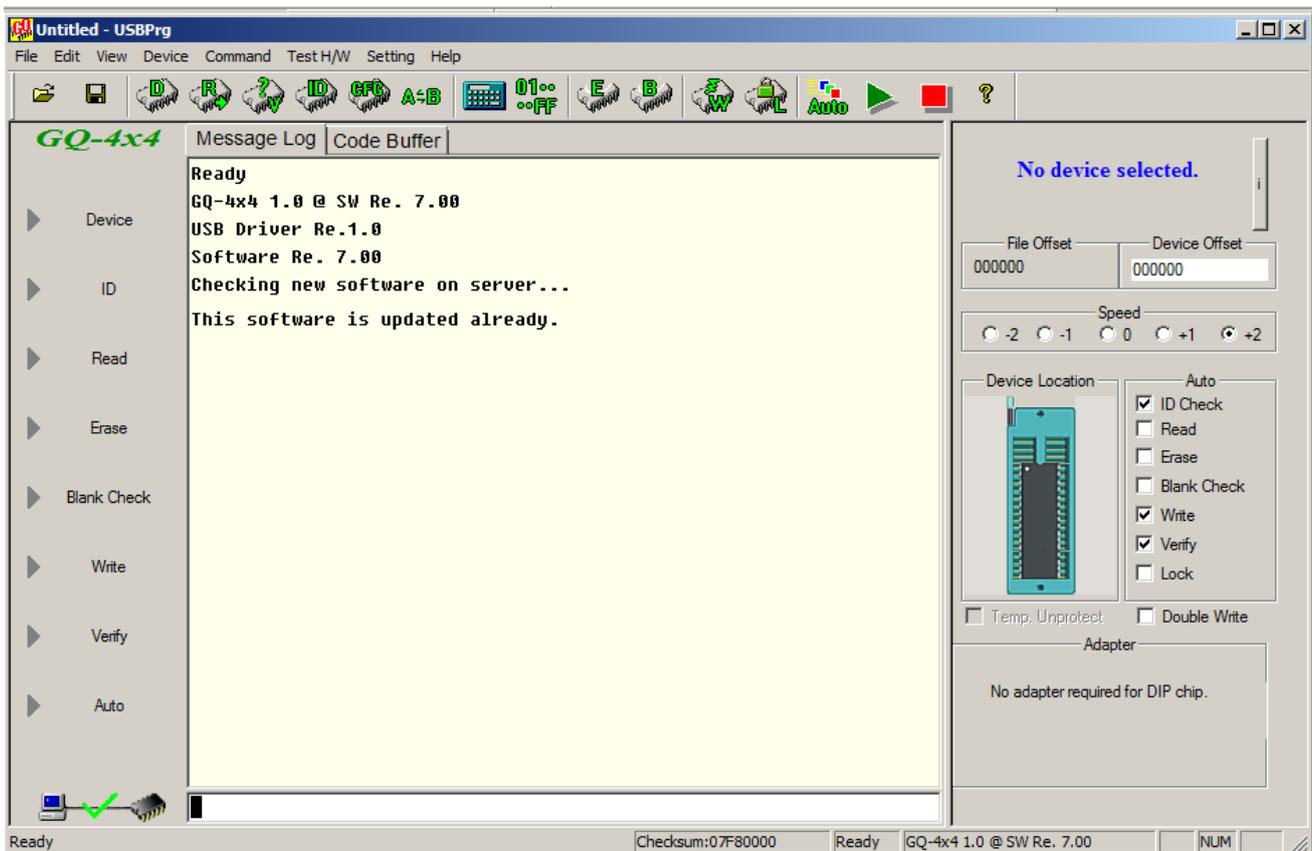


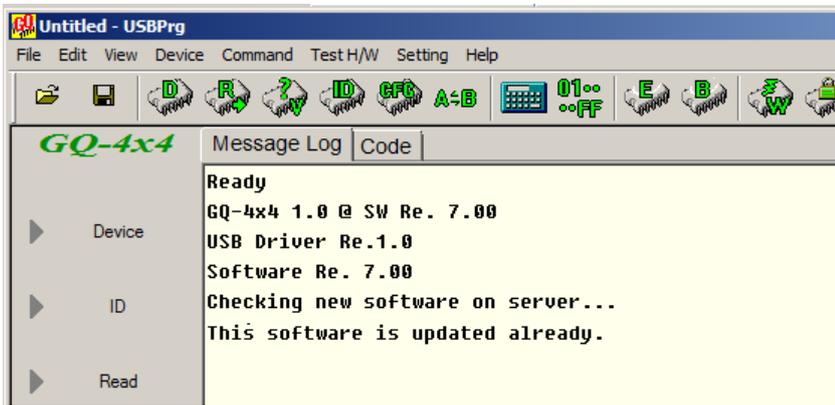
The USB cable is AB type Grade A cable. It should handle 500mA current for programmer. Always use our certified USB cable to ensure the programmer's functionalities.

The programmer external power supply is for those users need to connect the programmer to an unpowered USB hub.

Note: Do not remove or insert the chip when the Vcc/Busy or Vpp is on.

4. Software Interface



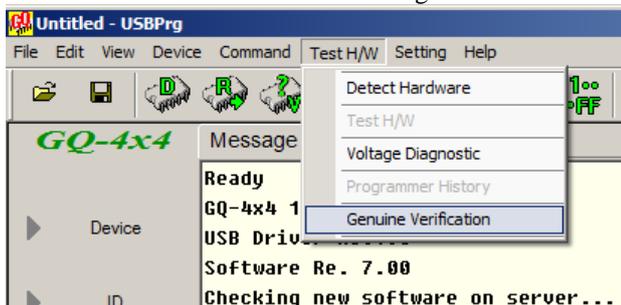


The main software user interface consists of the following areas:

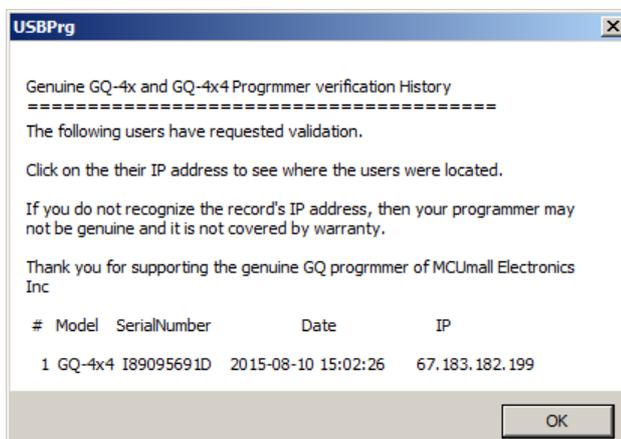
1. Left side: Command shortcut area
2. Middle area: message log and buffer area tabs
3. Right side: configuration panel
4. Bottom edge: status bar
5. Top edge: menu items and control tool bar
6. Message board providing user tips and messages for certain operations with specific chips

4.1 Genuine Programmer Verification

The software comes with Genuine Programmer Verification function.



Select this menu item to verify the programmer is genuine. None genuine programmer is not under warranty and will not work correctly. The below page shows the history of verification history.



Click on OK is going to open a webpage:

Genuine GQ-4x and GQ-4x4 Programmer Verification History

The following users have requested validation.

Click on the their IP address to see where the users were located.

If you do not recognize the record's IP address, then your programmer may not be genuine and it is not covered by warranty.

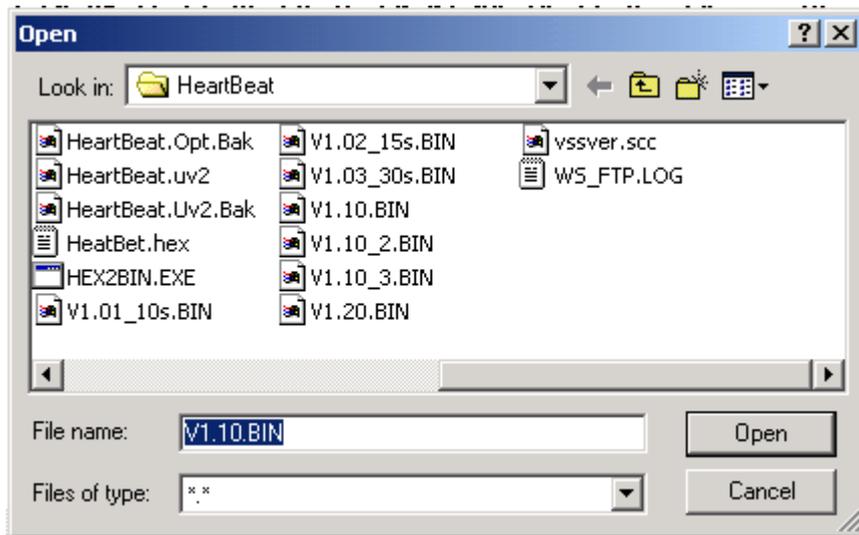
#	Model	SerialNumber	Date	IP
1	GQ-4x4	I89095991D	2015-08-10 15:02:26	68.183.182.199

Click on the IP address should open an IP-to-location page online. Check the location to see if anyone else own a same serial number programmer, which is a fake programmer.

4.1 Basic Menu and the Tool bar item



4.1.1 Load / Save data file



The following file formats are supported:

- Binary file: .bin
- Intel Hex file: .hex
- Motorola format: .s

Before loading the data file, you may want to change the file buffer offset (start address) from the configuration panel:



Once the data file has been loaded into the buffer, its contents will be shown in the data buffer window.

4.1.2 Modify the data buffer

Move the cursor to the position you want to change, click the mouse then key in the data in hexadecimal format.

4.1.3 Set the data buffer starting offset address in the buffer window

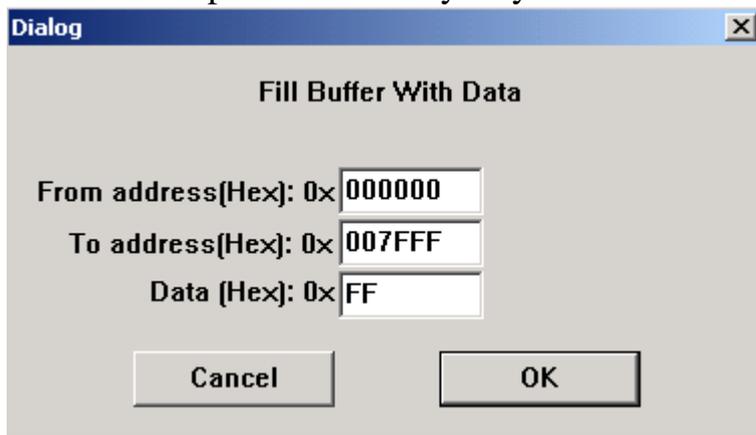
If a big data block is loaded into buffer, you may want to change the buffer starting address in the window. This is done by dragging the address buffer slider bar, so that you can easily see all other part data you wanted.

4.1.4 "Edit" menu item



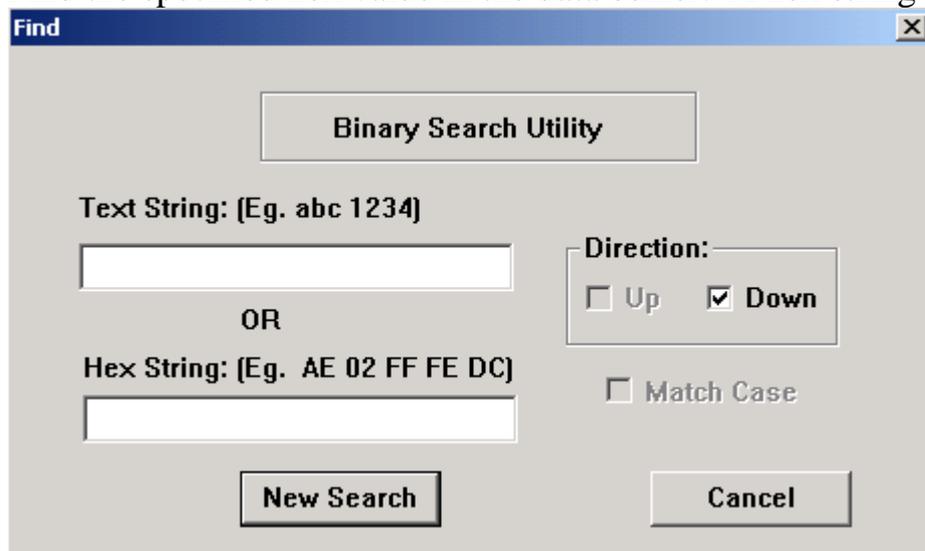
4.1.5 Fill Buffer

"Fill Buffer" provides an easy way to fill a buffer block with specified data.



4.1.6 "Find"

Find the specified hex value in the data buffer. A hex string is acceptable too.



4.1.7 The pop-up menu

If you right-click on the data area, a pop-up menu appears. You can copy/paste hex data from/to the buffer.

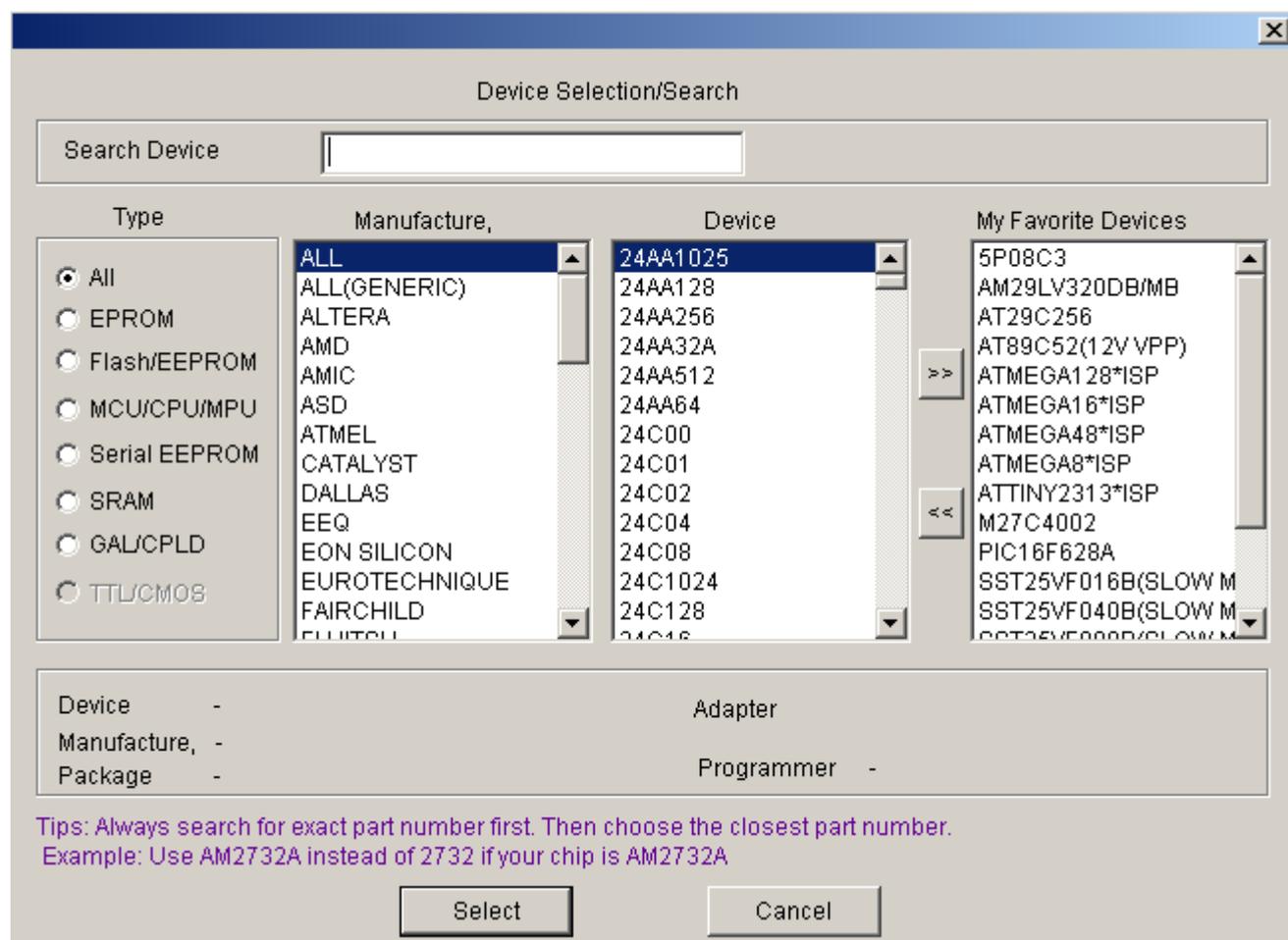
```

50 E3 66 00 00 0A-FE
0F E1          Copy  E9-C0
BD E8          Paste E8-FE
2D E9

```

4.2 Device type selection

When you click on the tool bar "D" button, this window will show up:



You can click on device name to select the device or highlight the device and then use the "Select" button to select it.

4.2.1 The My Favorite Devices List

You can use the ">>" and "<<" buttons to add or remove frequently used devices to speed up device selection.

4.2.2 Searching a devices from the device list

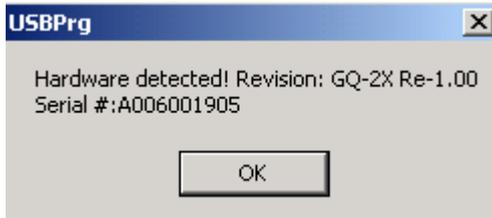
Type in the device name; the device list will be filtered by the string you've entered.

4.3 Test H/W menu item

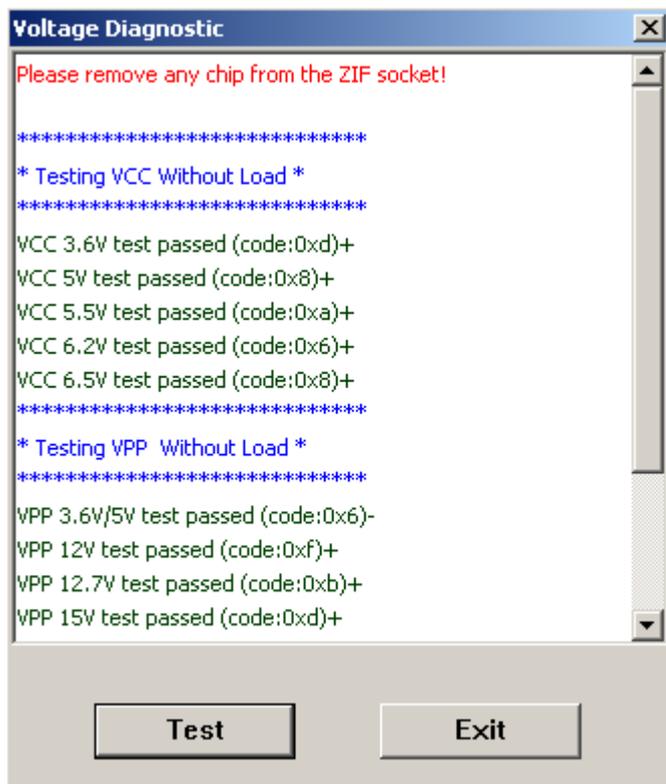
The software will test the system link status when it starts up. However you can test and detect the programmer at any idle time by clicking on the Test H/W menu item.



The message box will report the hardware revision and the hardware serial number.

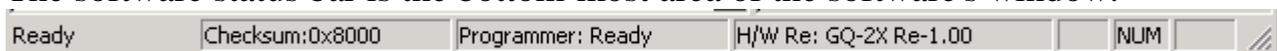


Test H/W Voltage menu item (GQ-4X Only) will automatically check all voltages available in system. The test result will be displayed in the Voltage Diagnostic message window.



4.4 Software status bar

The software status bar is the bottom-most area of the software's window.



It includes Prompt messages, Checksum, Programmer status and the Hardware version.

4.4.1 Checksum

The checksum will be calculated after the buffer has been loaded from a file or device.

4.4.2 Programmer status

The programmer's status will always be displayed on the status bar. A new operation will only be accepted when the programmer is idle.

4.4.3 Programmer revision

The programmer's hardware revision will be read from the hardware during hardware initialization phases.

4.5 Data Buffer Tabs

Different buffers may be displayed based on the device type by clicking one of the tabs at the top of the window. Here is example of a PIC16F84 device.



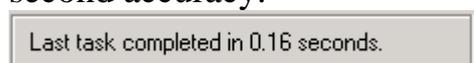
4.6 DIP setting indication (GQ-2X only)

The different device may have different DIP-switch setting. The software will pop up a message box when it needs to be set or checked.



4.7 Operation Speed Measurement

Once an operation task is completed, the duration used will be displayed with 1% second accuracy.



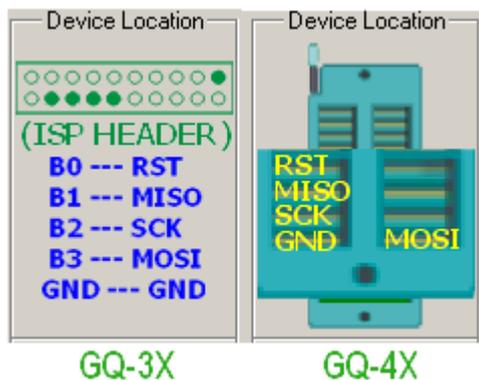
4.8 The adapters

The adapter information will be displayed after a device has been selected. It is possible that more than one adapter is available for a device.



4.9 The AVR ISP programming

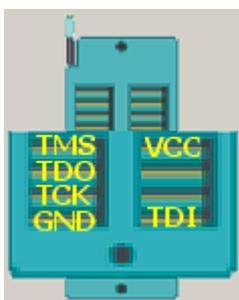
Both GQ-4x4 and GQ-4X support AVR ISP programming. The connection diagram indicated the connection pin-out to a standard Atmel AVR ISP port on the target board. The recommended connection wire length is about 20cm. Longer wire connections may cause unstable signals.



The AVR ISP connection pin-out.

4.10 The JTAG programming

The GQ-4X,GQ-4x4 supports the device JTAG programming. A JTAG programming driver adapter is recommended for the JTAG programming. User may use the JTAG signal from the ZIF socket directly, but the connection wire length may be limited depend on the design of target board.

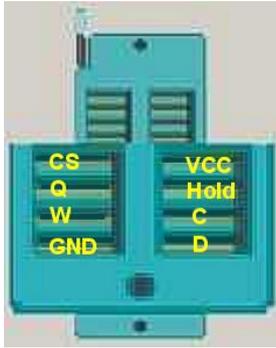


GQ-4X JTAG ZIF pin-out.

4.20 The SPI in-circuit programming

The GQ-4X,GQ-4x4 supports the SPI in-circuit-programming. A ADP-067 adapter is

recommended for the SPI in-circuit programming. User may use the SPI signal from the ZIF socket directly, but the connection wire length may be limited depend on the design of target board.

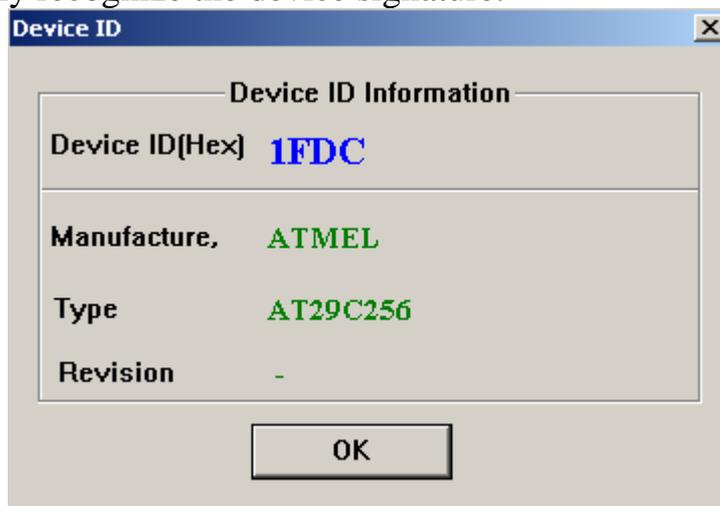


5. Device Operation

5.1 Steps

Connect the GQ USB programmer to the PC. Run programmer software.

1. Load the data file to buffer.
2. Confirm that the programmer supports the device, especially the EPROM's Vpp.
3. Lock the target device into the programmer's ZIF socket.
4. Select the device type.
5. Click on the ID button of tool bar. The programmer and the software should correctly recognize the device signature.



Example:

6. The Read/Write/Erase/Verify operation can now proceed.

5.2 File offset

files start at address 0, but you can specify any address as a start to write address. This should be set before Normally loading the data file into buffer. The default is 0.



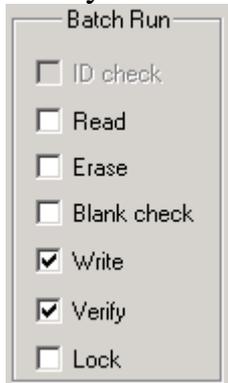
5.3 Device offset

This is the device start address. The specified address will be used for writing data as the starting address of the device.

5.4 Write (Program)

Before clicking on Write you **MUST** make sure the device type is correct and the programming voltage is correct(on some old EPROM only). Otherwise the target device maybe damaged.

The write operation includes a verify operation. It will prompt with an error message if writing fail. The extra verify operation will be executed if the write operation is completed successfully. However, this extra verification step can be skipped if the Verify check box is unchecked in the Batch Run item box.



5.5 *Read*

Read the device content into the program's buffer.

5.6 *Erase*

Erase the entire device and fill it with "FF". This only works on Electrically-Erasable devices. EPROMs need a separate UV eraser to erase the device.

5.7 *Blank*

Blank checks the device to see if it's empty or blank. This is an important step before any write operation. All bytes in the device must be filled "FF". It will stop the check and display a message if the device is not blank.

5.8 *Verify*

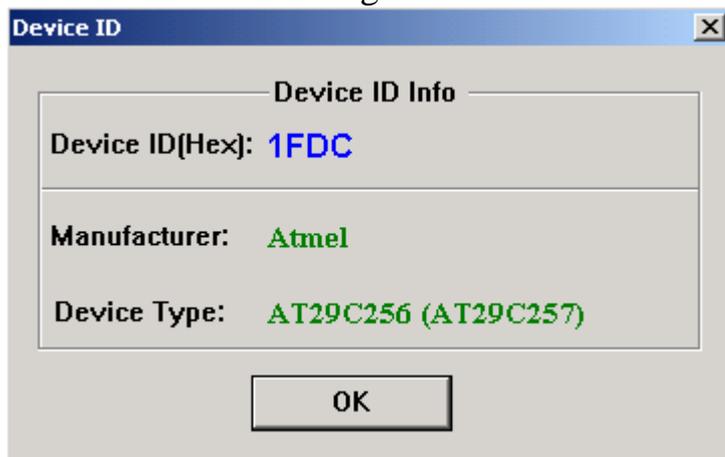
This compares the data in the buffer with the data in the device. It will stop the verification operation and display a message if a difference is detected.

5.9 Lock/Protect

Lock/Protect runs the lock utility feature. It will bring up the lock utility window based on the device selected.

5.10 Read ID

This reads the device signature. It includes manufacturer ID as well.



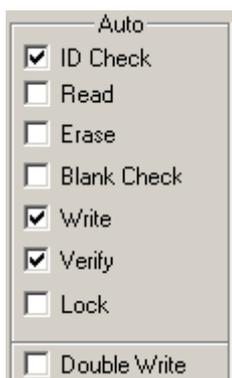
5.11 Speed

Different devices have different writing speed requirements, even if the device came from the same manufacturer. The speed setting is on the configuration panel. You can select five different speeds for all operation.



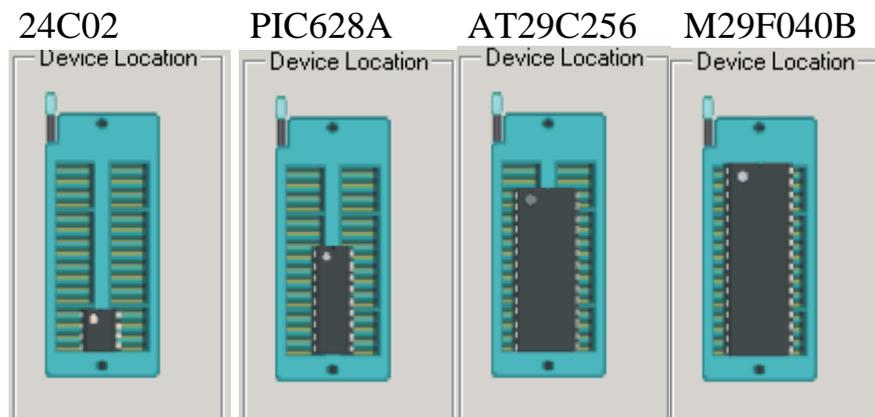
5.12 Auto Mode

The right side of the window shows the Auto operation list. Check the items you want to do all at one time then press the  button on toolbar.



5.13 Device Location in ZIF

The device location picture will be shown once the device has been selected. Here are the some examples of device locations:



6. The devices.txt file

The devices.txt file is the main supported device list configuration file. It is located in the application's directory. Advanced users may change it or add new devices to it.

The file format is text format and it can be opened and edited by any text editor. Each device has to be defined with one text entry, which can span several lines. The following device attributes can be defined individually:

```
Name="Device Name",  
ID="Device ID in Hexadecimal",  
Class="Class Name",  
Category="Category Type",  
MFG="MFG Name",  
Speed="Default Speed",  
Package="Package",  
Adapter="Adapter",  
CodeSize="Code Size",  
EepromSize="Eeprom Size",  
DIP="DIP Setting",  
VCC="Vcc",  
WVPP="Write Vpp",  
WVCC="Write Vcc",  
BVCC="Blank Vcc",  
RVPP="Read Vpp",  
Message="Text to be displayed to the user about switch settings, adapters, etc";
```

The 'Name=' and 'ID=' must be the first two fields and they must always be defined. All of the other fields are optional and are separated with commas. Long entries can be broken up into multiple short lines. If the first two fields are the only ones defined, then only the device ID will be added. The device itself will not be added to the search list. At least three fields are needed for any device entry; otherwise it will be ignored. All entries start with 'Name=' and end with a semicolon (;).

ID="XXXX" defines an unknown device ID or a generic device.

Comment lines start with a semicolon. Do not use the double quote as the first character in your comment. You can also put comments at the end of any device entry after the semicolon. Here is a typical line from the file:

```
Name="27C256" ,ID="298C" ,Class="27C256" ,Category="EPROM" ,MFG="Microchip" ;
```