GQ USB Universal Programmer

User Guide

Revision 4.11

Spt 2015

Revision History and Acknowledgement

Re 4.11 Revised for new graphic in software by Sept 2015
Re 4.10 Revised for new software by MCUmall Nov 2009
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

Special Thanks:

Robert W. Meister Revised English grammar and phrases.
Jorge Miguel Preto Provided chip location graphic design.

1. Table of Contents

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-2X, GQ-3X, GQ-4X) 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 98SE / ME / 2000 / XP / VISTA (both 32-bit and 64-bit).

IMPORTANT NOTE:

Please Install The Software Before You
Connect The Programmer To The Computer.
USB driver 1.0 Will Be Installed Automatically.

2. Software Installation

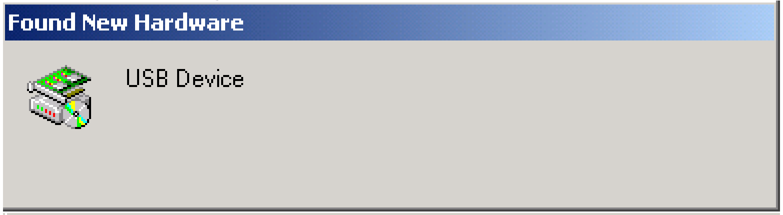
The software needs to be installed before you connect the programmer to a PC. This will eliminate the USB driver installation step on all 32-bit Windows system.

2.1 Windows 98SE / ME / 2000 / XP / Vista 32-bit

1. Unplug the programmer’s USB cable from the PC.
2. Run Setup.exe from CD and follow the installation instructions. Setup.exe will install both the application and a USB driver for the programmer.
3. Once the installation has completed and the PC has been restarted, connect the GQ USB programmer to the PC's USB port.

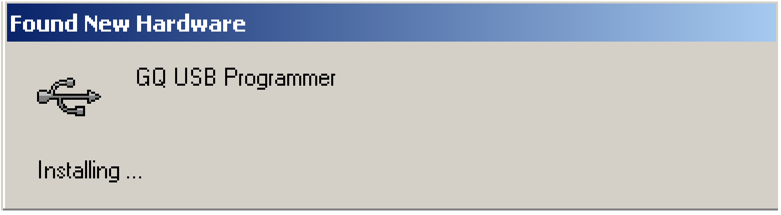
Note: If you are using an un-powered USB hub or extension hub, it may not provide enough power to operate the USB programmer. In this case, you may need a 9V, 200mA (or greater) DC adapter with a center-positive, 2.1mm DC plug.

For Win2000 you will see these installation screens:



Found New Hardware

USB Device



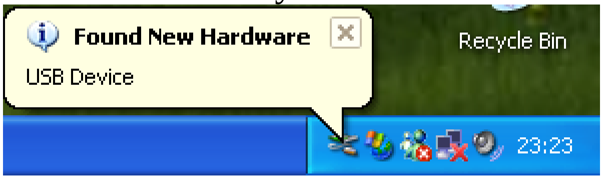
Found New Hardware

GQ USB Programmer

Installing ...

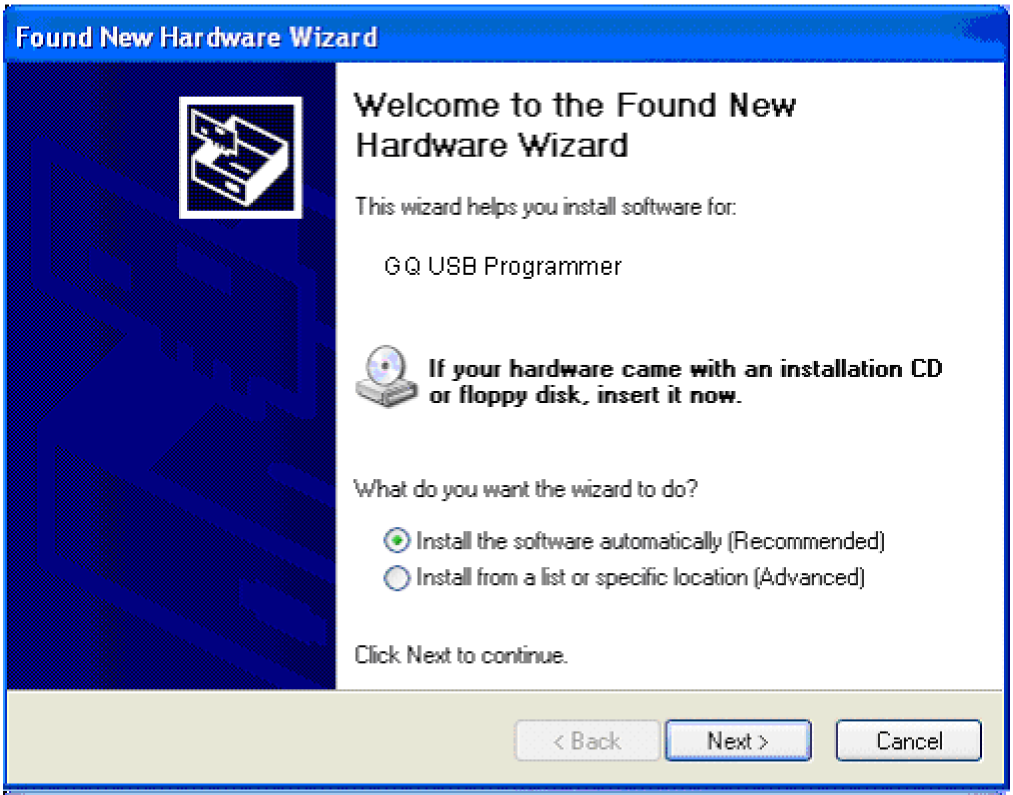
4. Restart the computer if it prompts you.
5. The programmer's green power LED should light up.

For Windows XP you will see these installation screens:



Found New Hardware

USB Device



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install software for:

GQ USB Programmer

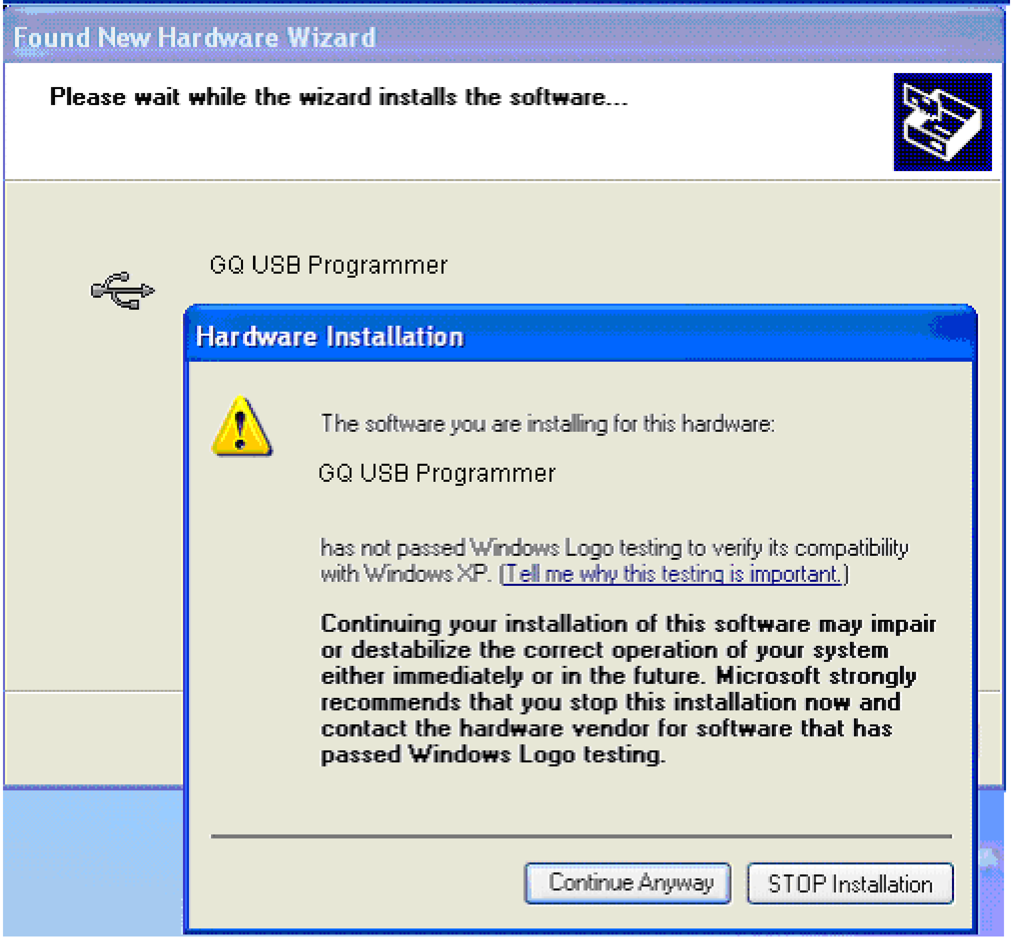
If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

● Install the software automatically (Recommended)

○ Install from a list or specific location (Advanced)

Click Next to continue.



Found New Hardware Wizard

Please wait while the wizard installs the software...

GQ USB Programmer

Hardware Installation

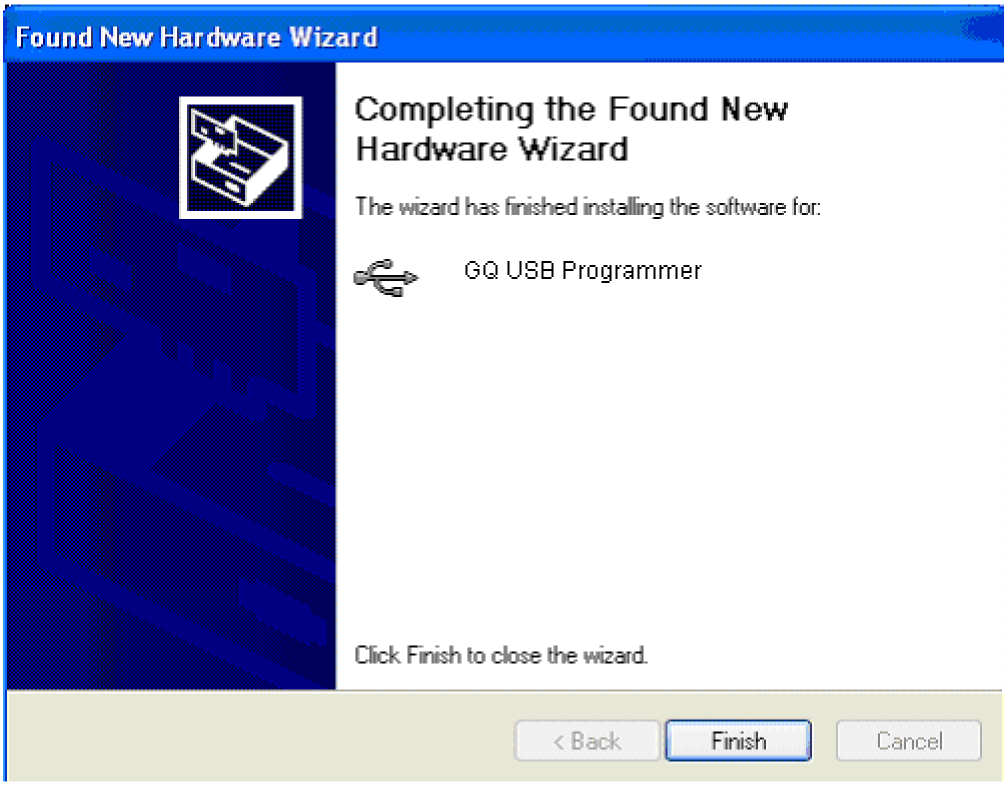
The software you are installing for this hardware:

GQ USB Programmer

has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.

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



Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

GQ USB Programmer

Click Finish to close the wizard.

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

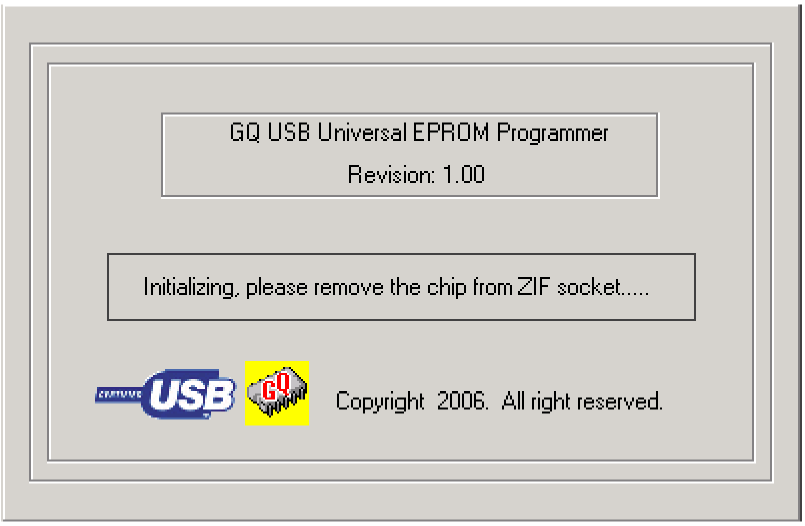
2.2 USB driver 2.0 or later version

The USB driver 2.0 or later version is a Microsoft digitally signed USB driver. It works on Windows XP SP2, XP 64-bit, Vista 32-bit and Vista 64-bit Windows systems.

The driver needs to be installed manually, either from Windows New Hardware Installation Wizard or Update the driver from the Windows Device Manager. The USB driver 2.0 has no impact on speed versus USB driver 1.0.

2.3 Start the programmer's software

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

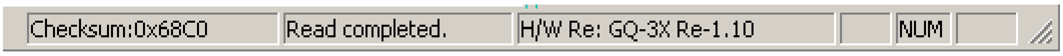


GQ USB Universal EPROM Programmer
Revision: 1.00

Initializing, please remove the chip from ZIF socket.....

Copyright 2006. All right reserved.

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

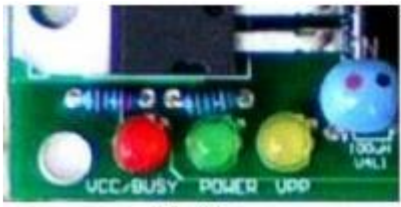


```
Checksum:0x68C0 | Read completed. | H/W Re: GQ-3X Re-1.10 | | NUM |
```

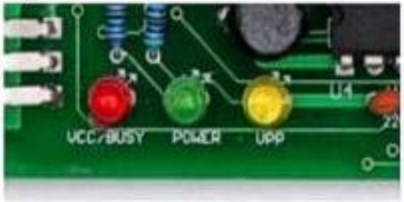
2. You are ready to program the device.

3. Hardware

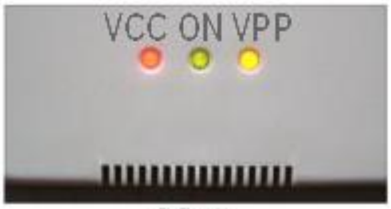
There are three LEDs located at the left bottom of programmer. Those LEDs indicate the current state of programmer.



GQ-2X



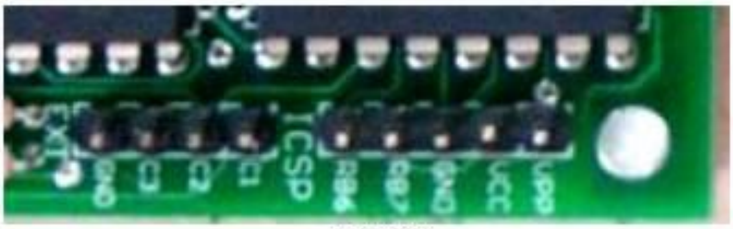
GQ-3X



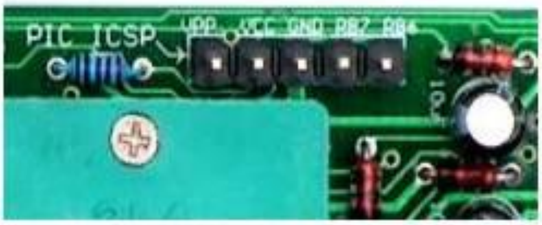
GQ-4X

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

1. The programmer board has an on-board 5-pin ICSP header. The ICSP header is located at the right edge side of the programmer. The ICSP header is for PIC chips that are not be able to fit in the ZIF.

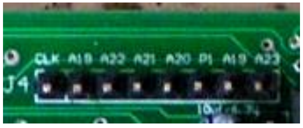


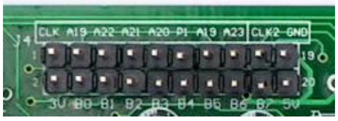
GQ-2X



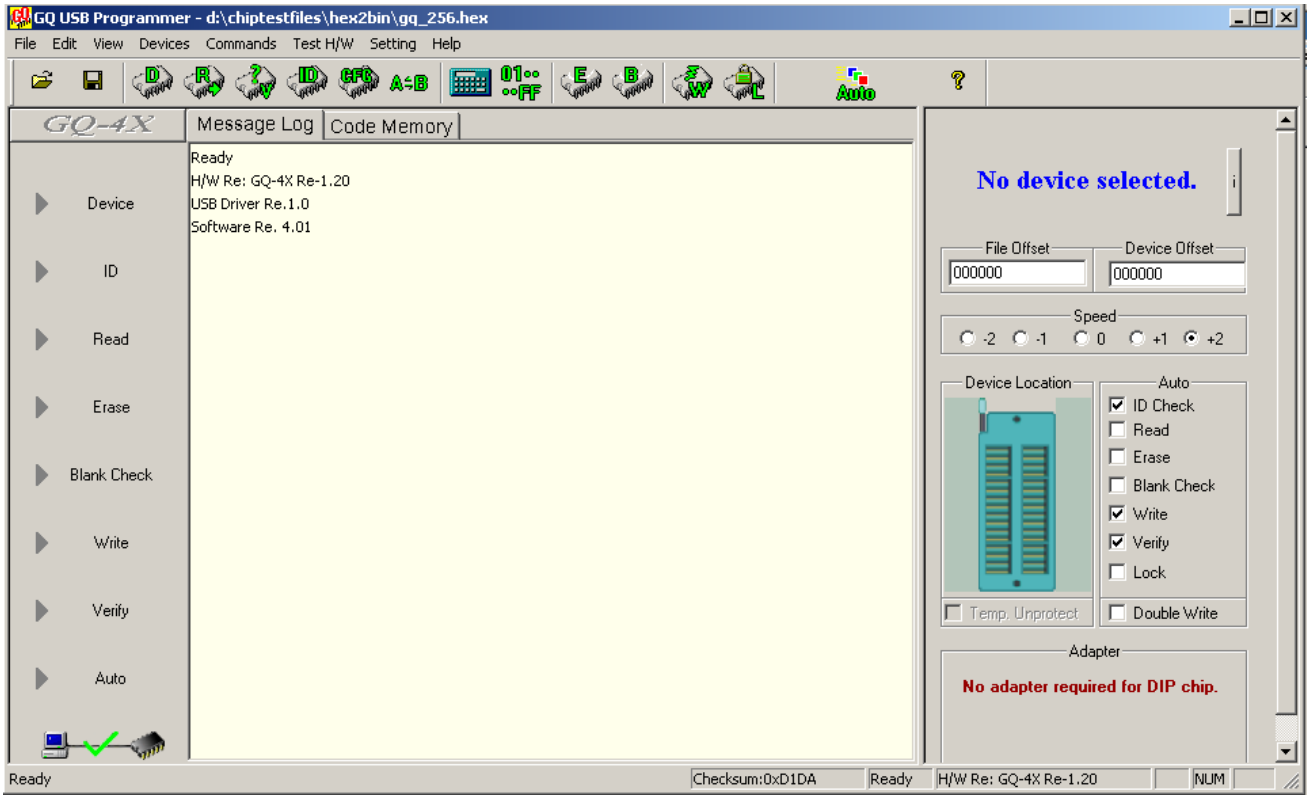
GQ-3X

2. The 4-pin EXT header is not used any more.
3. The DIP-switch on GQ-2X needs to be set before doing any operation. The software will show the proper switch pattern. The GQ-3X and GQ-4X do not have a DIP-switch.
4. The extended address line header is for devices that need the A19-A23, such as high-density flash chips. Some adapters will use this header too.





4. Software Interface


GQ USB Programmer - d:\chiptestfiles\hex2bin\gq_256.hex
File Edit View Devices Commands Test H/W Setting Help
Auto
GQ-4X
Message Log
Code Memory
Device
ID
Read
Erase
Blank Check
Write
Verify
Auto

```
Ready
H/W Re: GQ-4X Re-1.20
USB Driver Re.1.0
Software Re. 4.01
```

No device selected.

File Offset: 000000
Device Offset: 000000

Speed: ○ -2 ○ -1 ○ 0 ○ +1 ● +2

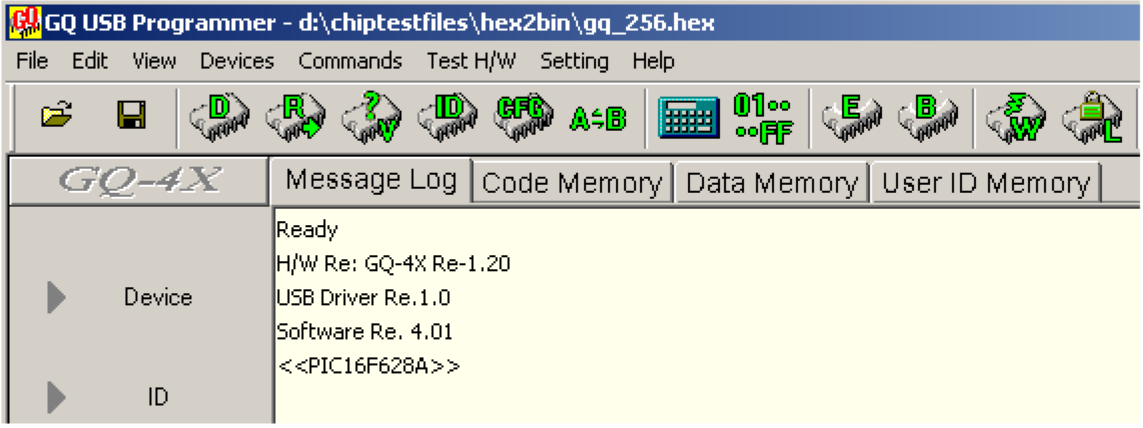
Device Location

Auto:
☑ ID Check
□ Read
□ Erase
□ Blank Check
☑ Write
☑ Verify
□ Lock

□ Temp. Unprotect
□ Double Write

Adapter: No adapter required for DIP chip.

Ready | Checksum:0xD1DA | Ready | H/W Re: GQ-4X Re-1.20 | NUM

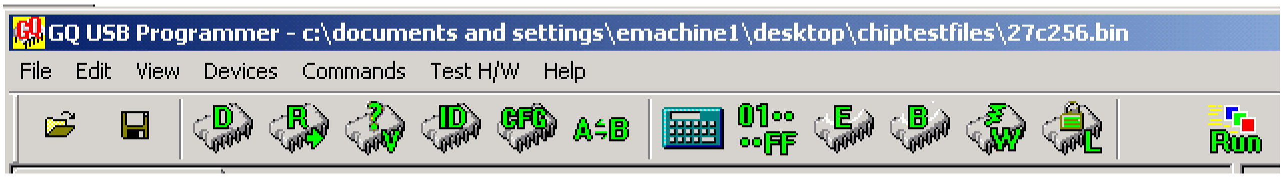

GQ USB Programmer - d:\chiptestfiles\hex2bin\gq_256.hex
File Edit View Devices Commands Test H/W Setting Help
GQ-4X
Message Log
Code Memory
Data Memory
User ID Memory
Device
ID

```
Ready
H/W Re: GQ-4X Re-1.20
USB Driver Re.1.0
Software Re. 4.01
<<PIC16F628A>>
```

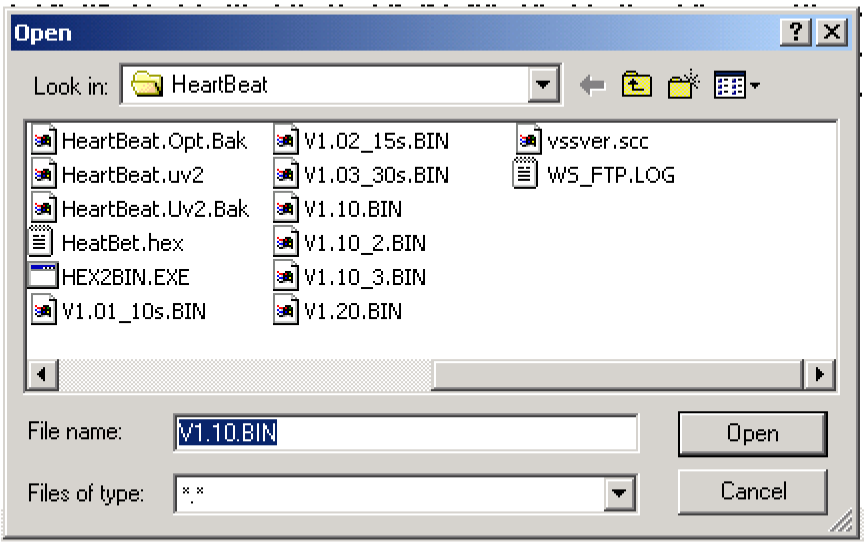
The main software user interface consists of the following areas:

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 Basic Menu and the Tool bar item


GQ USB Programmer - c:\documents and settings\emachine1\desktop\chiptestfiles\27c256.bin

4.1.1 Load / Save data file


Open

Look in: HeartBeat

HeartBeat.Opt.Bak
HeartBeat.uv2
HeartBeat.Uv2.Bak
HeatBet.hex
HEX2BIN.EXE
V1.01_10s.BIN
V1.02_15s.BIN
V1.03_30s.BIN
V1.10.BIN
V1.10_2.BIN
V1.10_3.BIN
V1.20.BIN
vssver.scc
WS_FTP.LOG

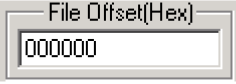
File name: V1.10.BIN

Files of type: *.*

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:


File Offset(Hex)
000000

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

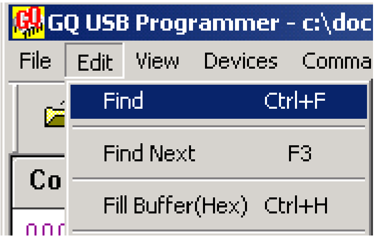
4.1.2 Modify the data buffer

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

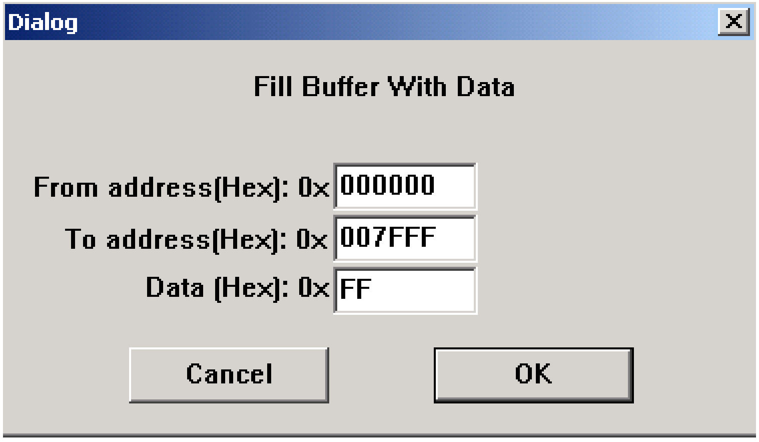
If a big data block is loaded into buffer, you may want to change the buffer starting address in the window. This is down by dragging the address buffer slider bar, so that you can easily see all other part data you wanted.

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.



Fill Buffer With Data

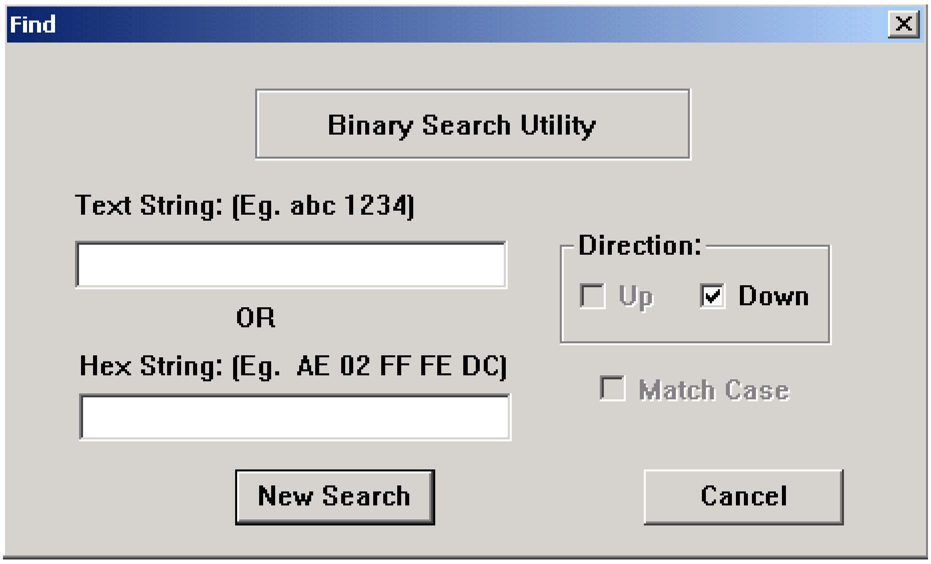
From address(Hex): 0x 000000

To address(Hex): 0x 007FFF

Data (Hex): 0x FF

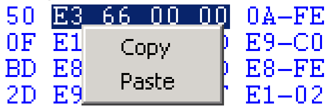
4.1.6 "Find"

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



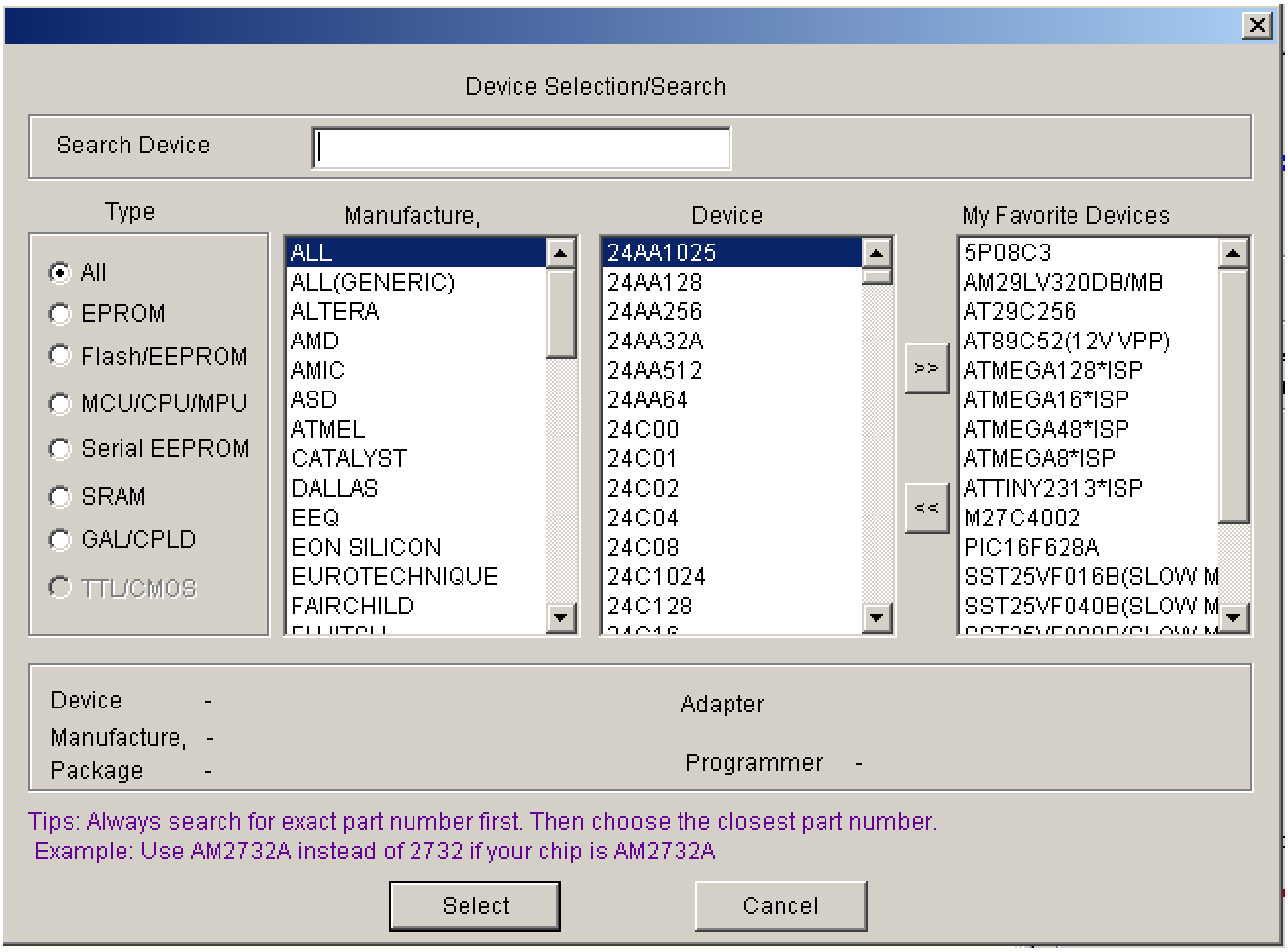
4.1.7 The pop-up menu

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              E9-C0
BD  E8              E8-FE
2D  E9              E1-02
```

4. 2 Device type selection

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



Type: ● All ○ EPROM ○ Flash/EEPROM ○ MCU/CPU/MPU ○ Serial EEPROM ○ SRAM ○ GAL/CPLD ○ TTL/CMOS

Manufacture,: ALL, ALL(GENERIC), ALTERA, AMD, AMIC, ASD, ATMEL, CATALYST, DALLAS, EEQ, EON SILICON, EUROTECHNIQUE, FAIRCHILD

Device: 24AA1025, 24AA128, 24AA256, 24AA32A, 24AA512, 24AA64, 24C00, 24C01, 24C02, 24C04, 24C08, 24C1024, 24C128

My Favorite Devices: 5P08C3, AM29LV320DB/MB, AT29C256, AT89C52(12V VPP), ATMEGA128*ISP, ATMEGA16*ISP, ATMEGA48*ISP, ATMEGA8*ISP, ATTINY2313*ISP, M27C4002, PIC16F628A, SST25VF016B(SLOW M, SST25VF040B(SLOW M

Device -
Manufacture, -
Package -

Adapter

Programmer -

Tips: Always search for exact part number first. Then choose the closest part number.
Example: Use AM2732A instead of 2732 if your chip is AM2732A

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

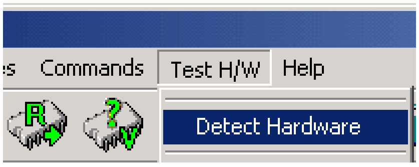
You can use the ">>" and "<<" buttons to add or remove frequently used devices to speed up device selection.

4.2.2 Searching a devices from the device list

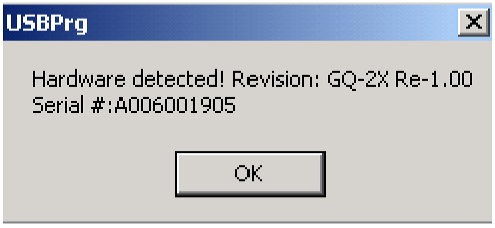
Type in the device name; the device list will be filtered by the string you've entered.

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.

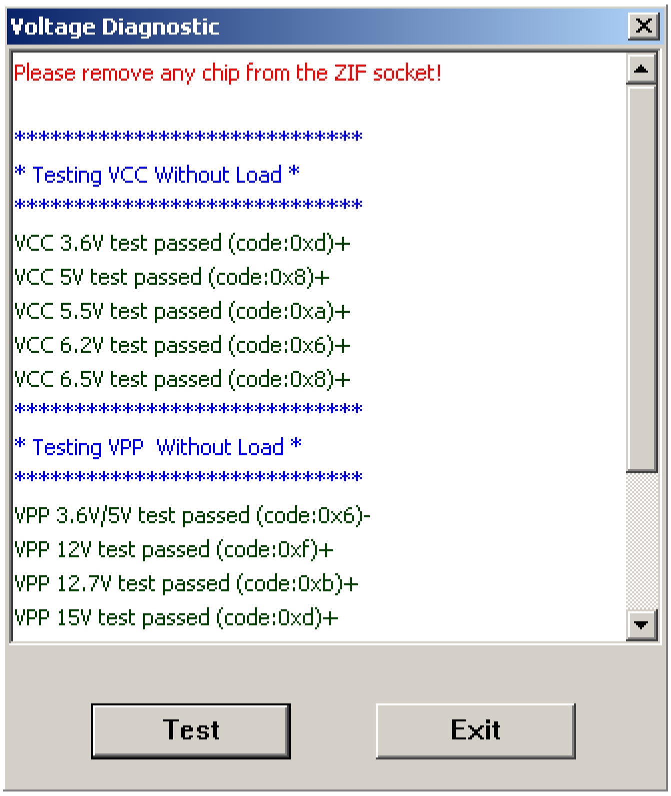


The message box will report the hardware revision and the hardware serial number.



Hardware detected! Revision: GQ-2X Re-1.00
Serial #:A006001905

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.

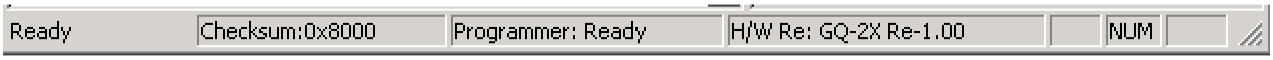


```
Please remove any chip from the ZIF socket!

******************************
* Testing VCC Without Load *
******************************
VCC 3.6V test passed (code:0xd)+
VCC 5V test passed (code:0x8)+
VCC 5.5V test passed (code:0xa)+
VCC 6.2V test passed (code:0x6)+
VCC 6.5V test passed (code:0x8)+
******************************
* Testing VPP  Without Load *
******************************
VPP 3.6V/5V test passed (code:0x6)-
VPP 12V test passed (code:0xf)+
VPP 12.7V test passed (code:0xb)+
VPP 15V test passed (code:0xd)+
```

4.4 Software status bar

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



Ready | Checksum:0x8000 | Programmer: Ready | H/W Re: GQ-2X Re-1.00 | NUM

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

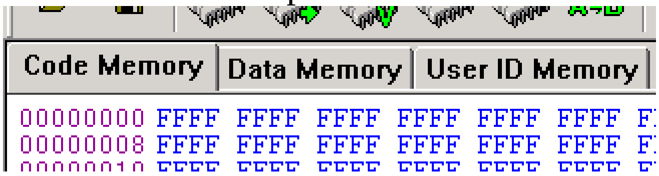
The programmer's status will always be displayed on the status bar. A new operation will only be accepted when the programmer is idle.

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

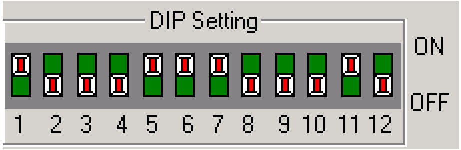
Different buffers may be displayed based on the device type by clicking one of the tabs at the top of the window. Here is example of a PIC16F84 device.



```
00000000 FFFF FFFF FFFF FFFF FFFF FFFF F
00000008 FFFF FFFF FFFF FFFF FFFF FFFF F
00000010 FFFF FFFF FFFF FFFF FFFF FFFF F
```

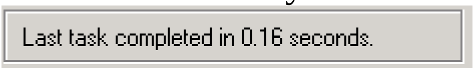
4.6 DIP setting indication (GQ-2X only)

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

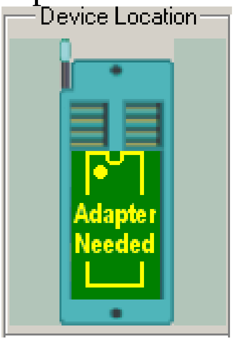
Once an operation task is completed, the duration used will be displayed with 1% second accuracy.



Last task completed in 0.16 seconds.

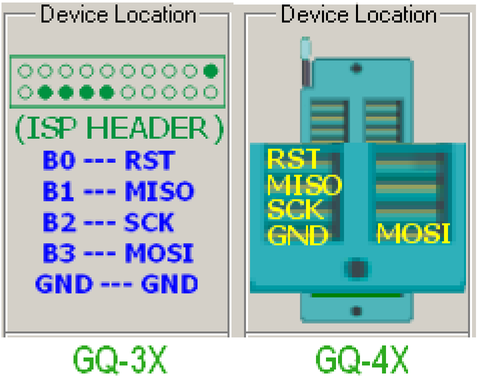
4.8 The adapters

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-3X and GQ-4X support AVR ISP programming. The AVR ISP connection information will be displayed differently on GQ-3X and GQ-4X. 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.

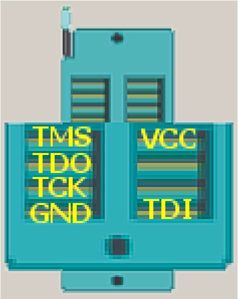


(ISP HEADER)
B0 --- RST
B1 --- MISO
B2 --- SCK
B3 --- MOSI
GND --- GND

The AVR ISP connection pin-out.

4.10 The JTAG programming

The GQ-4X 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.

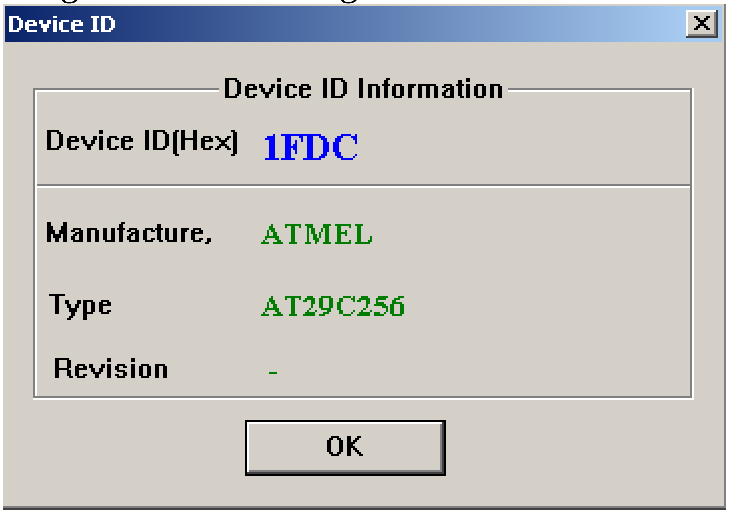
5. Device Operation

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:



Device ID Information

Device ID(Hex) 1FDC

Manufacture, ATMEL

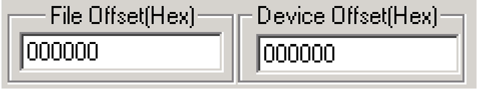
Type AT29C256

Revision -

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.



File Offset(Hex) 000000

Device Offset(Hex) 000000

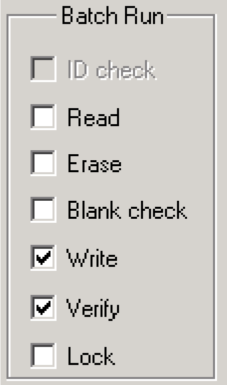
5.3 Device offset

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.



Batch Run

□ ID check

□ Read

□ Erase

□ Blank check

☑ Write

☑ Verify

□ Lock

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

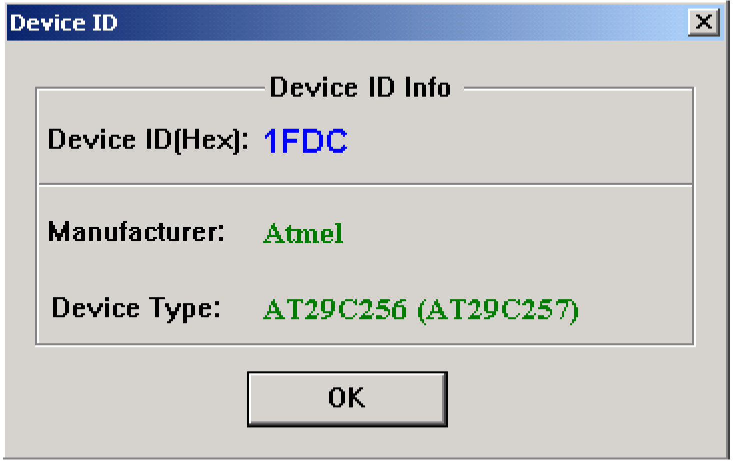
This compares the data in the buffer with the data in the device. It will stop the verification operation and display a message if a difference is detected.

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.



Device ID

Device ID Info

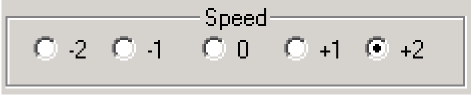
Device ID(Hex): 1FDC

Manufacturer: Atmel

Device Type: AT29C256 (AT29C257)

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.

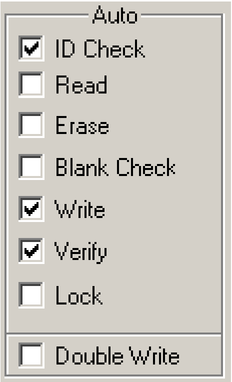


Speed

○ -2 ○ -1 ○ 0 ○ +1 ● +2

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 Auto button on toolbar.



Auto

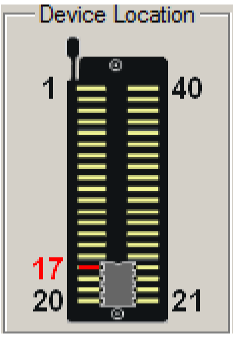
☑ ID Check
□ Read
□ Erase
□ Blank Check
☑ Write
☑ Verify
□ Lock

□ Double Write

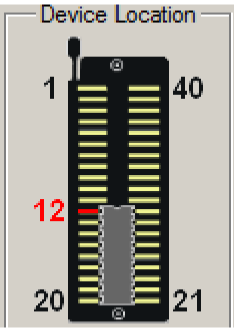
5.13 Device Location in ZIF

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

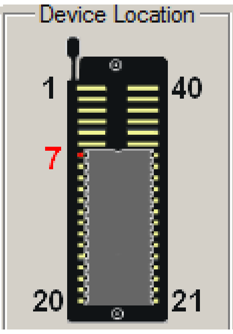
24C02



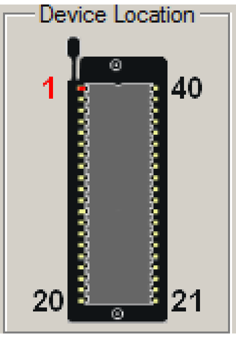
PIC628A



AT29C256



M29F040B



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 (;).

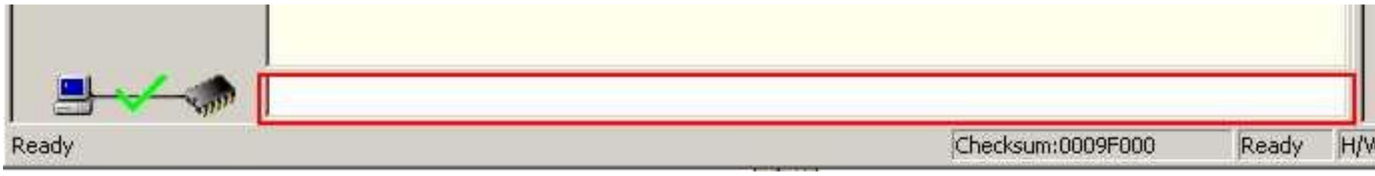
ID="XXXX" defines an unknown device ID or a generic device.

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";
```

7. Command line operation

The software is able to accept operation command from software command line box. User may type the operation command from keyboard. The latest command set will be listed when type the “?” or “help” in the command line box.



The following are example commands (Capital letters are key word):

? list all command available
BLANKCHECK
BYTESWAP
CHECKBOX id/read/write/erase/blankcheck/double write/temp protect (check the checkbox)
CLS (clear main message log screen)
COLOR LOG BKG ffffff (where ffffff is the RGB color digit)
COLOR COMMAND BKG ffffff (where ffffff is the RGB color digit)
DEVICE SST27SF512
ERASE
EXIT APP (exit the programmer software from windows)

FILL <data>
FILL <data> <fromAddress(hex)>
FILL <data> <fromAddress(hex>) <toAddress(hex)>
GOTO <label>
HELP (this help message)
LABEL <only>
LOAD <myData.bin>
LOG <this is a message log only.>
MESSAGE <my message prompt>
OFFSET FILE <offset value>
OFFSET DEVICE <offset value>
READ
SCRIPT <myScript.gqs>
SPEED <speed>(eg. +2)
STOP (stop the script)
UNCHECKBOX <id/read/write/erase/blankcheck/double write/temp protect> (uncheck the checkbox)
VERIFY
WAIT <seconds> (wait seconds)
WRITE

7.1 DOS prompt command line

The GQ programmer software is able to run from DOS command prompt with a GQScript file as optional parameter.

Example #1:

```
C:\GQUSBPrg.exe
```

This will lunch the programmer software.

Example #2:

```
C:\GQUSBPrg.exe StartScript.gqs
```

This line will lunch the programmer software and then run the script file “StartScript.gqs” automatically once the software and programmer initialized.

8. The GQScript file .gqs

GQScript is a unique feature of GQ USB Programmer. It allows user to edit a programming sequences and steps , and then run it automatically. It uses same command set of command line. The file extension is .gqs

The comment line start with ";"

The scrip will stop running when an error occurred, such as verify failed, write failed etc. The script will prompt to get a response if ID check failed. <BR><BR>

There are GQScript example files located in GQScript directory.

```
;//////////////////////////////////////

;    GQScript Demo

;//////////////////////////////////////

LABEL AT29C256

message Please be ready for AT29C256

color log bkg f0ff0f

device at29c256

load D:\ChipTestFiles\my29C256Data.bin

fill ff 00B1 00B1

checkbox ID

uncheckbox verify

wait 1

erase

blankcheck

write

verify

uncheckbox verify

color log bkg ff0f0f

message Please be ready for W27E257
```

```
log ***********************

device W27E257

log ***********************

read

verify

erase

blankcheck

write

verify

color log bkg ffffeb

goto AT29C256

message Script End
```