# UG01 USB to GPIB Controller

## USER'S MANUAL



Documentation ID: 004001

Version: 1.0

Issue Date: 11/13/2016

LQ Electronics Corp

# UG01 USB to GPIB Controller USER'S MANUAL
# TABLE OF CONTENTS

## 1.0    GENERAL INFORMATION

## 1.1    UG01 USB to GPIB Controller and Software Overview

**UG01 USB to GPIB Controller** is powerful, friendly and low cost GPIB controller. It is very easy for use. The software includes ASCII format command/data transfer functions and binary data/ file transfer functions between PC and GPIB equipment. We already compact everything you need for driving the equipment with GPIB interface. Therefore, you can eliminate the complicated VISA library.

The software for UG01 USB to GPIB Controller is UG01 Software, which includes three main elements:
1) One is the UG01 control panel (UG01.exe), which is user friendly Graphic User Interface (GUI). User can control GPIB interface equipment by inputting GPIB command and read data or test result from the GPIB equipment by just clicking the Data read button after you sent GPIB query command.
2) Second is Application Programmable Interface (API) for cdecl calling convention, which includes static library (LQUG01_c.lib) and dynamic -link library (LQUG01_c.dll). Users can write your own programs to realize remote control of the GPIB interface equipment in any languages except VB/VBA.
3) Third is Application Programmable Interface (API) for stdcall calling convention, which includes static library (LQUG01_s.lib) and dynamic -link library (LQUG01_s.dll). This is for VB and VBA clients since VB/VBA only take stdcall calling convention. Users can write your own programs to realize remote control of the GPIB interface equipment, even in Windows Office software, such as Excel (VBA).

The API examples have been installed with the software together. The examples are built by Visual Studio 2005 at 32bit OS of Windows XP with SP3.

UG01 Software has been tested on 32bit OS of Windows XP with SP3, 32bit OS of Windows 7 with SP1, 64bit OS of Windows Vista with SP1, 64bit OS of Windows 7 with SP1, 32bit OS of Windows 10 and 64bit OS of Windows 10. For Windows 10 user, please download .NET Framework 3.5 from Microsoft official website and install it if .NET Framework 3.5 has not installed on your system.

Detailed functioning of the software will be covered in Section 2 in this document.

For UG01 and UGSimple USB to GPIB Controller comparison, please see table below.

**Table 1    UG01 and UGSimple USB to GPIB Controller Comparison**

| Model | ASCII character length for command/data | Binary data transfer | Screen image Capture | GPIB script feature | File read and save | GPIB equipment drives |
|-------|------------------------------------------|----------------------|----------------------|---------------------|--------------------|------------------------|
| UGSimple | 60 | No | No | No | No | <5 |
| UG01 | 20000 | Yes | Yes | Yes | Yes | 14 |

The screen image capture on the UG01 Control Panel works for GPIB IEEE488.2 and does not work for HPIB IEEE488.1. If you want to capture a screen image for HPIB interface, you have to write a program with binary read function ( *Gbread* or *Gfilesave* ) to realize a screen image capture. See sample program of UG01_dll_VC++ or UG01_dll_Excel.

## 1.2   Authorized Use Permission

Usage of this software is limited to its owner via the terms of its development. UG01 Software is wholly owned by LQ Electronics Corp, and may not be used or referenced without their express consent.

## 1.3   Points of Contact

### 1.3.1  Information

For additional information, please email: sales@lqelectronics.com.

## 2.0   UG01 SOFTWARE

## 2.1   UG01 GUI

UG01 GUI (UG01.exe) is installed in the folder of C:\Program Files\LQElectronics\UG01\UG01GUI for 32bit Windows OS, or in the folder of C:\Program Files (x86)\LQElectroncis\UG01\UG01GUI for 64bit Windows OS. It is a user friendly GUI and very easy for user to control UG01 USB to GPIB Controller.

UG01 USB to GPIB Controller control Panel
If you plug an UG01 USB-GPIB Controller on your computer, you will see the Icon as shown in Fig.1 when you run UG01.exe.



Fig. 1 UG01 Control Panel

**2.1.1 Functions of each item**
Fig.2 illustrates the items which are shown in the Fig.1.

The functions of each item is explained as below,
1.  Button "Find"
Click this button to find all the GPIB interface equipment connected on the GPIB bus.
2.  GPIB address window for GPIB equipment initialization
Input GPIB address or click the GPIB address shown in item 12 for the equipment you want to initialize (see item 14).
3.  GPIB address window for GPIB equipment communication
Input GPIB address or click the GPIB address shown in item 12 for the equipment you want to connect to.
4.  GPIB command window
Input GPIB command you want to send to specific equipment
5.  Browse button
Click this button to find a specific GPIB script for loading.
6.  GPIB script file path and file name window
After clicking Browse button, the GPIB script file path and name for loading will be shown in the window.
7.  Information window
All information will be shown in this window
8.  Load button
Click the button to load GPIB script. The GPIB script will be loaded and run after the button clicked.
9.  Data read button
Read test result or data from a equipment after you sent query command
10. GPIB address window for Capture Screen button
Input GPIB address or click the GPIB address shown in item 12 for the Capture Screen button
11. Capture Screen Button
Click the button to capture the screen image for the equipment of GPIB address shown in item 10.
12. Display GPIB addresses for the equipment connected on the GPIB bus
All the GPIB addresses of the equipment connected on the GPIB bus are displayed here after clicking "Find" button or when control panel is launched.
13. GPIB command write button
Writing GPIB command to specific equipment, GPIB address is shown in GPIB address window.
14. Initialize button
Click the button to initialize specific GPIB equipment.

## 2.1.2. Using UG01 Control Panel
Using the UG01 Control Panel
- Find equipment connected on the GPIB bus
  By clicking "Find" button, the GPIB addresses of the equipment will be shown at item 12.
  Click a GPIB address at item 12, the address will be shown in item 2, item 3 and item 10 so as to avoid input GPIB address at GPIB address window (item 2, item 3 and item 10).
- Initialize a GPIB equipment (for IEEE488.2 only)
  Click the button to initialize specific GPIB equipment: Clicking the button will send a GPIB common command "*RST" to the GPIB equipment, which address is shown in item 2.
- Write GPIB command to equipment
  Input or click the GPIB address at item 12, and input GPIB command at item 4, then click "Command Write" button
- Read data or result from equipment.
  Click "Data read" button to read data or result from the equipment, after you sent query command. The data read is ASCII format.

- Capture screen image – please see **Note**
  If you want to capture the screen image of the GPIB equipment if necessary, you can click "Capture Screen" button. It will prompt a folder on local PC for where to save the screen image then save the screen image as GIF file. This will take around 3 ~ 6 minutes to read the binary file from the GPIB equipment and save it.
  **Note**:   "Capture screen" works on most of equipment with IEEE488.2 GPIB, and does not wok on equipment with IEEE488.1 –HPIB.
- Load a GPIB script
  If you want to load a GPIB script, you can click "Browse" button, and find the GPIB script. Then click "Load" button, the GPIB script will be loaded. If there are some GPIB query commands, the result read from GPIB equipment will be saved at the folder/file path the dame with the script. The saved file name is the name of the GPIB script plus "_result.txt".
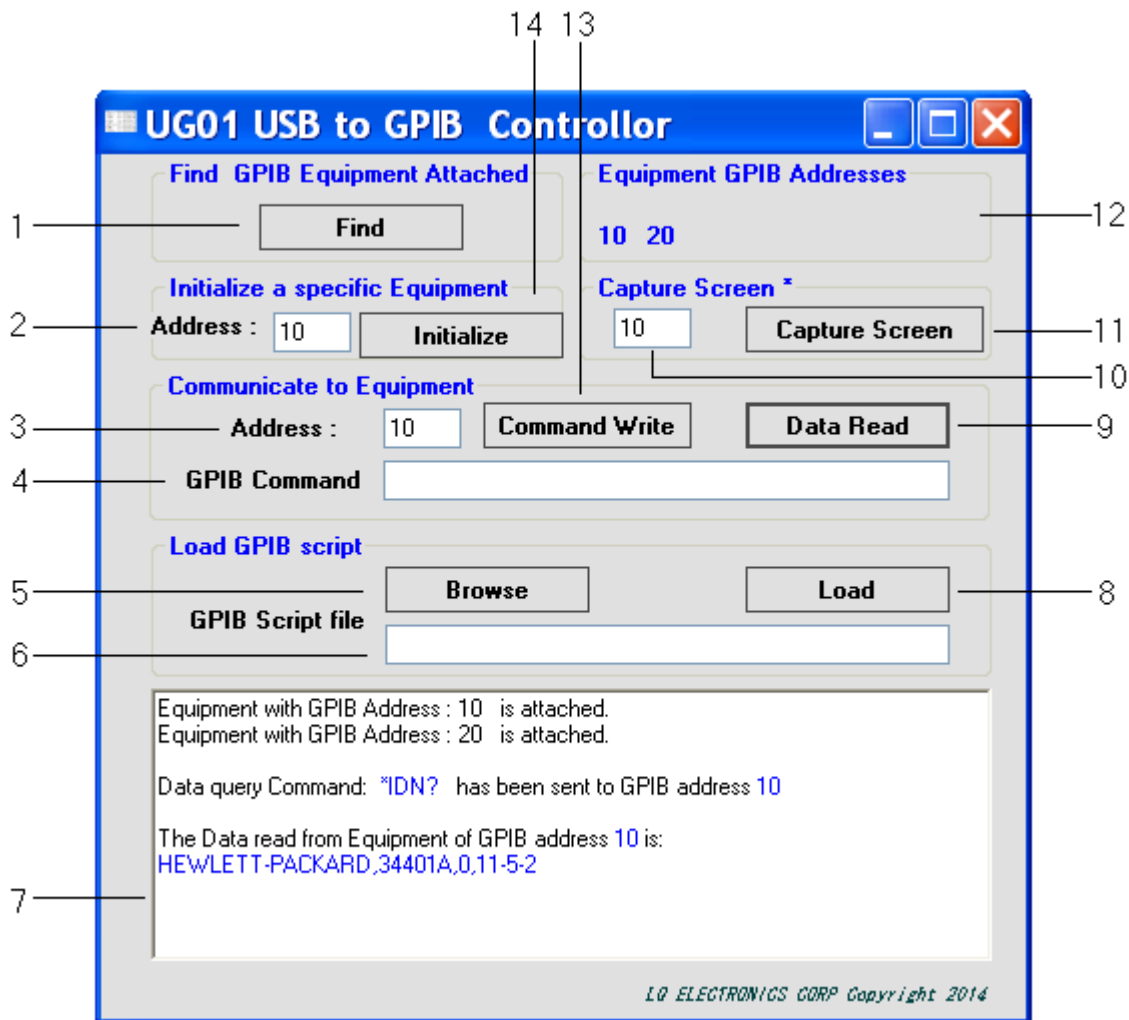


Fig.2 Items appeared on UG01 Control Panel

### 2.1.3. GPIB script

The GPIB script is for the convenience of the people who is not familiar with program languages. The script is a text file, which stores the GPIB equipment address and the GPIB command, which can be obtained from user manual of most equipment. Just follow the defined format, a person who doesn't

know program languages can realize remote control of GPIB equipment and get the data from the equipment.

The script format is very simple,
First letter of a line should be "a" or "A": means following is GPIB address after the letter. After the GPIB address is dash "-" for isolating the address and GPIB command. After dash "-", there is GPIB command to be sent. If the command to be sent is a command for query (command contains "?"), the data or result read will be saved to ****_result.txt. Where **** is the name of this script. If a line is started by ";", this line will be treated as comment, and will not be executed.

The script sample is at folder C:\Program Files\LQElectronics\UG01\UG01API\Samplefiles\GPIBscript for 32bit Windows OS. For 64bit Windows OS, it is at folder C:\Program Files (x86)\LQElectronics\UG01\UG01API\Samplefiles\GPIBscript.

GPIB script can only be loaded by UG01 Control Panel and cannot be called by any functions.

## 2.2   UG01 Software API - cdecl

UG01 Software API – cdecl is comprised of header file LQUG01_c.h, static library LQUG01_c.lib and dynamic-link library LQUG01_c.dll. The files are installed in the folder of C:\Program Files\LQElectronics\UG01\UG01API for 32bit Windows OS or C:\Program Files (x86)\LQElectronics\UG01\UG01API for 64bit Windows OS. The static library and dynamic library are written of Visual Studio 2005 C++ and can be called by user's programs in any languages (except VB/VBA) at Windows OS, such as C/C++, Python. It is very convenient to call the functions in the libraries.

UG01 API -cdecl functions

Function A: Write GPIB command to specific equipment
*int* Gwrite (*int*   address, *char* * scpi);

The input variables of the function are integer *address* and character pointer *scpi,* where store the GPIB address of specific equipment and GPIB command respectively. The return value should be 0 if function is successful. Otherwise, the return value is -1.

Function B: Read data or test result from equipment
*char* * Gread(*int*   address);

The input variable is integer *address,* which stores the address of the equipment you want to read after you wrote query command to the equipment. The return value is the character pointer of the data (ASCII characters) read back if it is successful, otherwise return decimal 10 (Hex 0xA). The maximum length of the ASCII characters read back is 20000.

If first character returned is Linefeed (0xA), maybe USB or GPIB equipment is not connected or GPIB address error.

Function C: Write binary data (or binary file) to specific equipment
*int* Gbwrite(*int*   address, *bool* mode,   *unsigned char* * bdata, *int* writelength );

There are four input variables. First variable is integer *address*, which stores the address of the equipment you want to write. Second variable is Boolean type *mode,* which stores the binary data sending mode. If *mode* is true or 1, the binary data (or file) will be sent in Definite Length format. Otherwise, the binary data will be sent in Indefinite Length format. Which format is chosen depends on the GPIB equipment requirement. Third variable is unsigned character pointer *bdata,* which stores the binary data (or file). Fourth variable is integer *writelength,* which stores the length to be sent of the binary data (or file). You must know the length of the binary data (or file) so that UG01 can encode the binary data to be Definite Length or Indefinite Length. The return value should be 0 if function is successful. Otherwise, the return value is -1.

Function D and E:   Read binary data (or file) from specific equipment and Get the received binary data (or file) length.
Function D: `Unsigned char` * Gbread(`int`   address);
Function E: `int` `Gbreadlength`(`void`);

Generally these two functions are used together. The input variable in Function D is integer *address,* which stores the address of the equipment you want to read after you wrote query command to the equipment. The return value is the unsigned character pointer of the data read back (Windows OS defined "unsigned char" as "BYTE") if function is successful, otherwise return decimal 10 (Hex 0xA). The maximum length of the unsigned characters read back is 100000.

Function E returns data length of the binary data (or file), which has been read by Function D. It has no variable. After Function D called, the return value of Function E has generated automatically.

For binary data (or file) reading, UG01 can detect the coding format (Definite Length or Indefinite length) automatically and decode the received binary data, then obtain the data length information. Therefore the returned data from Function D is decoded binary data (or file data). The length coding has been eliminated.

Function F: Read a file from specific GPIB equipment and save it to local PC
`int` Gfilesave(`int`   address, `bool` mode, `char` *   filename);

There are three variables in Function F. First variable is *address*, which stores the address of the GPIB equipment for reading. Second is Boolean type *mode,* which stores the mode of the file for reading. Set variable *mode* to be true or 1 if the file is binary file, such as screen image, trace data file(CSV file), trace status file (.sta) or other file type unknown; Set variable *mode* to be false or 0 for reading a text file. Third is a character pointer *filename,* which is stores the name and path of the file to be saved on the local PC. This name can be the same or different with file name on the GPIB equipment.

Function G: Find the addresses of the GPIB equipment connected on the GPIB bus
`Int`* Gfind (`void`);

There is no variable in Function G. The return value is integer pointer for the addresses of the GPIB equipment connected on the GPIB bus after Function G is called.

Function H: Query function. This function is simply combined of *Gwrite* and *Gread*.
`Char`* Gquery (`int`   address, `char` * scpi);

The function is a read-after-write function. After it wrote GPIB command to the GPIB equipment, it read the result back. The variable of integer *address* stores the address of the GPIB equipment. The variable of character pointer *scpi* stores the GPIB command. The return value is the character pointer of the data read back (ASCII characters) if the function is successful, otherwise return decimal 10 (Hex 0xA). The maximum length of the ASCII characters read back is 20000.

Function I: Function to get Firmware version
*Double GetFWVersion (void);*

The function has no variable. The return value is a double value as #.# if the function is successful, such as 1.0 for Firmware version 1.0. Otherwise the return value is -1. The function can be used to check if the UG01 is plugged into USB socket of a PC. If not plugged, return value is -1, otherwise UG01 will return the firmware version.

Function J: Suppress all warning messages
*int Setwarningmessage (bool    setwm);*

The function is to set not to display warning messages or other messages in all functions. The variable *setwm* in the function is Boolean type. If *setwm* is true or 1, all messages in the functions will be displayed when necessary. Otherwise, all messages will be suppressed. The default value of the messages display is true. The function should be called before other functions.

Function I and Function J are for software developers or product developers who would like to integrate UG01 into their products or into their projects.


## 2.3    UG01 API - stdcall

UG01 API – stdcall is comprised of header file LQUG01_s.h, static library LQUG01_s.lib and dynamic-link library LQUG01_s.dll. The files are installed in the folder of C:\Program Files\LQElectronics\UG01\UG01API for 32bit Windows OS or C:\Program Files (x86)\LQElectronics\UG01\UG01API for 64bit Windows OS. The static library and dynamic library are written of Visual Studio 2005 C++ and can be called by user's programs in any languages at Windows OS, such as VB/VBA or C/C++. It is very convenient to call the functions in the libraries.

UG01 API -stdcall functions

Function A: Write GPIB command to specific equipment
*Int _stdcall  Gwrite(int   address, char * scpi);*

The input variables of the function are integer *address* and character pointer *scpi,* where store the GPIB address of specific equipment and GPIB command respectively. The return value should be 0 if function is successful. Otherwise, the return value is -1.

Function B: Read data or test result from equipment
*char *_stdcall  Gread(int   address);*

The input variable is integer *address,* which stores the address of the equipment you want to read after you wrote query command to the equipment. The return value is the character pointer of the data (ASCII

characters) read back if it is successful, otherwise return decimal 10 (Hex 0xA). The maximum length of the ASCII characters read back is 20000.

If first character returned is Linefeed (0xA), maybe USB or GPIB equipment is not connected or GPIB address error.

Function C: Write binary data (or binary file) to specific equipment
*int _stdcall Gbwrite(int address, bool mode, unsigned char * bdata,, int writelength );*

There are four input variables. First variable is integer *address*, which stores the address of the equipment you want to write. Second variable is Boolean type *mode,* which stores the binary data sending mode. If *mode* is true or 1, the binary data (or file) will be sent in Definite Length format. Otherwise, the binary data will be sent in Indefinite Length format. Which format is chosen depends on the GPIB equipment. Third variable is unsigned character pointer *bdata,* which stores the binary data (or file). Fourth variable is integer *writelength,* which stores the length to be sent of the binary data (or file). You must know the length of the binary data (or file) so that UG01 can encode the binary data to be Definite Length or Indefinite Length. The return value should be 0 if function is successful. Otherwise, the return value is -1.

Function D and E:   Read binary data (or file) from specific equipment and Get the received binary data (or file) length.
Function D:   *Unsigned char * _stdcall  Gbread(int  address);*
Function E:   *int _stdcall  Gbreadlength(void);*

Generally these two functions are used together.  The input variable in Function D is integer *address,* which stores the address of the equipment you want to read after you wrote query command to the equipment. The return value is the unsigned character pointer of the data read back (Windows OS defined "unsigned char" as "BYTE") if function is successful, otherwise return decimal 10 (Hex 0xA). The maximum length of the unsigned characters read back is 100000.

Function E returns data length of the binary data (or file), which has been read by Function D. It has no variable. After Function D called, the return value of Function E has generated automatically.

For binary data (or file) reading, UG01 can detect the coding format (Definite Length or Indefinite length) automatically and decode the received binary data, then obtain the data length information. Therefore the returned data from Function D is decoded binary data (or file data). The length coding has been eliminated.

Function F: Read a file from specific GPIB equipment and save it to local PC
*int _stdcall  Gfilesave(int  address, bool mode, char * filename);*

There are three variables in Function F. First variable is *address*, which stores the address of the GPIB equipment for reading. Second is Boolean type *mode,* which stores the mode of the file for reading. Set variable *mode* to be true or 1 if the file is binary file, such as screen image, trace data file(CSV file), trace status file (.sta) or other file type unknown; Set variable *mode* to be false or 0 for reading a text file. Third is a character pointer *filename,* which is stores the name and path of the file to be saved on the local PC. This name can be the same or different with file name on the GPIB equipment.

Function G: Find the addresses of the GPIB equipment connected on the GPIB bus
*Int* * *_stdcall  Gfind (void);*

There is no variable in Function G. The return value is integer pointer for the addresses of the GPIB equipment connected on the GPIB bus after Function G is called. If error is found, the returned address becomes -1 or -2.

Function H: Query function. This function is simply combined of *Gwrite* and *Gread*.
*Char*\* *_stdcall*   *Gquery (int*   *address, char \* scpi);*

The function is a read-after-write function. After it wrote GPIB command to the GPIB equipment, it read the result back. The variable of integer *address* stores the address of the GPIB equipment. The variable of character pointer *scpi* stores the GPIB command. The return value is the character pointer of the data read back (ASCII characters) if the function is successful, otherwise return decimal 10 (Hex 0xA). The maximum length of the ASCII characters read back is 20000.

Function I: Function to get Firmware version
*Double _stdcall*   *GetFWVersion (void);*

The function has no variable. The return value is a double value as #.# if the function is successful, such as 1.0 for Firmware version 1.0. Otherwise the return value is -1. The function can be used to check if the UG01 is plugged into USB socket of a PC. If not plugged, return value is -1, otherwise UG01 will return the firmware version.

Function J: Suppress all warning messages
*int _stdcall*   *Setwarningmessage (bool*   *setwm);*

The function is to set not to display warning messages or other messages in all functions. The variable *setwm* in the function is Boolean type. If *setwm* is true or 1, all messages in the functions will be displayed when necessary. Otherwise, all messages will be suppressed. The default value of the messages display is true . The function should be called before other functions.

Function I and Function J are for software developers or product developers who would like to integrate UG01 into their products or into their projects.


## 2.4   Visual Basic (VB) and Visual Basic Application (VBA) Modules

UG01VB32.bas, UG01VB32.vb, UG01VB64.bas and UG01VB64.vb are VB modules.

UG01VB32.bas is for VBA on 32bit Window OS and UG01VB64.bas is for VBA on 64bit Windows OS. When you write a VBA program, you can add or import the module to your program so as to call the functions from LQUG01_s.dll.

UG01VB32.vb is for VB in Visual studio 2005 or other version Visual studio on 32bit Windows OS and UG01VB64.vb is for VB in Visual studio 2005 or other version Visual Studio on 64bit Windows OS respectively. When you write a VB program, you can add the module to your program so as to call the functions from LQUG01_s.dll.

Function Gfind in VB and VBA module
The return value of Gfind in VB and VBA module is not integer pointer. It is string. Each ASCII character represents an address of the GPIB equipment. The true integer address of the GPIB equipment is the ASCII character value minus 0x30.
Example 1
If the ASCII character in the returned string is "9", then the true GPIB address is 9, which is equal to 0x39(ASCI I value of character "9") minus 0x30.
Example 2
If the ASCII character in the returned string is "A", then the true GPIB address is 17, which is equal to 0x41(ASCCII value of character "A") minus 0x30

## 2.5   UG01 API vs. LabView

It is possible for UG01 API –cdecl to be loaded by LabView because LabView is a C compiler and Functions in library of LQUG01_c.dll are exported by standard C format. All function is exported with extern "C". For detailed information of calling functions of third party library into LabView, please refer to link at: http://www.ni.com/white-paper/3341/en/#toc5 or http://www.ni.com/tutorial/3009/en/

## 3.0   UG01 HARDWARE

The USB interface of UG01 USB-GPIB Controller is full speed USB 2.0. UG01 USB to GPIB Controller is a full function of GPIB Controller. It can drive 14 GPIB interface equipment, which comply with IEEE488.2 and IEEE488.1 requirement. Only one UG01 USB-GPIB Controller can be installed at a computer. For product details, Please visit http://www.lqelectroncis.com/products/USBUG/USBUG.html to compare both models.

## 4.0    GETTING STARTED

Installing UG01 Software and starting Application

## 4.1    Installing UG01 Software

UG01 Software is properly compressed in an installation file, which is called UG01_v##_setup.exe. The ## is a two digits version of the software.

All the files are included in UG01_v##_setup.exe. UG01_v##_setup.exe is stored on a CDR, which is attached with the UG01 USB to GPIB Controller when you buy the product. In addition, you can download it from website: http://www.lqelectronics.com/support/USBUG/download.html

When you start to install the UG01 USB to GPIB Controller, follow the steps below,

1.  Plug the UG01 USB to GPIB Controller to a computer; ignore all reminders of the computer.
2.  Insert the CDR to DVD/CD drive, double-click the file of UG01_v##_setup.exe
3.  Follow the instruction appeared on the screen of your monitor, the software will be installed automatically.

**We strongly recommend you plug UG01 USB to GPIB controller to the computer before install UG01 Software to avoid window assigns wrong USB driver.**

## 4.2    Starting the Application

The application of UG01 GUI is installed into the startup program menu at LQElectronics\UG01\. Just click it, the UG01 GUI will start up.

**LQ Electronics Corp**

Tel: (408)836-8112
Email: sales@lqelectronics.com
http://www.lqelectronics.com

March 27, 2015