TEXIO
Test and Measurement Solutions

INSTRUCTION MANUAL

For IF-60RU / 70GU / 80GUR

Windows API and USB device driver

Ver.3.10

＜Licensing Agreement for the Software＞

1. The consent of the right

TEXIO TECHNOLOGY CORPORATION (hereunder TEXIO TECHNOLOGY) consents the below mentioned right to the customers who agree this licensing Agreement for this downloadable software and the related materials (hereunder the SOFTWARE).

(a) The customers can use the SOFTWARE for the use of our products corresponding to the SOFTWARE.
(b) The customers can duplicate the SOFTWARE for the operation of plural number of computers.

2. Additional consent article

The various files made as the result of the use of the SOFTWARE with the agreed purpose shall be the customer's literary work.

3. Copyright

Copyright of the SOFTWARE and the duplicate shall be hold by TEXIO TECHNOLOGY and the person whom TEXIO TECHNOLOGY admitted and is protected by Japanese Copyright Act and a Global Treaty. When the customers make duplicate of the SOFTWARE under this Licensing Agreement for the SOFTWARE, the same copyright indication of the SOFTWARE downloaded shall be required.

4. Prohibition

When the SOFTWARE is supplied as source code, the customers shall not distribute it revising source code to the third party as TEXIO TECHNOLOGY made.

5. No Warranty

TEXIO TECHNOLOGY applies no warranty on suitability for the customers' specific purpose or utility, or no defects in the SOFTWARE, or all other matters related to the SOFTWARE.

6. Exemption from responsibility

TEXIO TECHNOLOGY does not accept any responsibility for whatever damages (damage of the business, the interruption of the business, loss of the business information, or the other financial damage) caused by use or nonuse of the SOFTWARE.

7. Cancellation of the agreement

TEXIO TECHNOLOGY can cancel the agreement when the customer violates the licensing agreement of the SOFTWARE The customer shall not use the SOFTWARE after the cancellation of the agreement.

＜Warranty coverage＞

Download and installation of the SOFTWARE shall be made by the customers with their own responsibility.
Revision or improvement of the SOFTWARE may be made without notice.

＜Copyright holder＞

Each copyright of the SOFTWARE belongs to TEXIO TECHNOLOGY.

＜Trademarks＞

Windows, Windows XP, Windows VISTA ,Windows7, Visual Basic, Visual Basic for Application, Visual C++ are registered as trademarks of Microsoft Corporation.

CONTENTS

1. OUTLINE

・ The SOFTWARE consists of a device driver for USB and a module(API) for access, and controls IF-60RU, IF-70GU and IF-80GUR from Windows PC.

2. FEATURES

・ Available header and sample files for various languages help easy programming.

3. PRODUCT REQUIREMENT

・ PC can be operated by Microsoft　XP SP3/VISTA(32bit),and Windows7/8(32bit/64bit)
・ Use the USB device driver made by TEXIO TECHNOLOGY and it adapts to USB1.1 and USB2.0 full speed.
・ Usable languages are VB6.0,VB2008/2010,VBA,VC#,and VC++ (C,C++).
・ If you want to use the GP-IB in the sample program. Use National Instruments Corp.
(The NI488.2M Software, please use the Ver.2.8.1 or more.)

4. INSTALLATION OF API

・ Please Login to PC by administrator.
・ Extract the downloaded file to the appropriate folder, and then run as an administrator by right-clicking Setup32.exe(32bit) / Setup64.exe(64bit) for your environment.
・ Run the. Exe Setup ** in the Driver folder in a row, please install the device driver if you are using a USB.
・ When using VBA and the VisualBasic6.0 (Excel2000 or more), you can use the API by adding to IF_ ***.Bas. When using VisualBasic2005 or higher, you can use the API by adding to IF_ ***.vb
・ When using the Visual C + +, you can use the API by adding a file to the project IF_ * 0.h, IF_ * 0API_Class.h, of IF_ * 0API_Class.cpp.
・ The usage of the API, please refer to the here and so have been prepared sample program.
・ Confirmation message to continue or set during installation will appear depending on the OS, but please continue as it is.
・ You may specify the folder appear during installation, the installation folder is not created, the file is copied to the system folder.
・. Installation of VC + + runtime library and netFramework4 library will be performed automatically. Run the TEXIO_API32.msi or TEXIO_API64.msi, If you can't install in competition with the system.
・ Depending on your environment, run the TEXIO_API32.msi or TEXIO_API64.msi reinstall time.

5. INSTALLATION OF USB DEVICE DRIVER

5-1.for Installation

・ Please connect to a PC that is running the USB cable while turn OFF the power switch of the main body.
・ Login to PC by administrator.
・ Please install and run as administrator by right-clicking Setup32.exe(32bit) / Setup64.exe(64bit) in the Driver folder. Message appears when you do not pass the test of the Microsoft logo in the middle, but please continue as it is.
・ Wait for the installation is complete, PC is to recognize by turning ON the power switch of the main body. Please note it may take time depending on the situation and the number of the connected equipment.
・ If it is recognized, the detection of new hardware begins. Select the installation automatically, install the device driver according to the message. Message would not have passed the test of the Microsoft logo in the middle comes out, but please continue as it is. Confirmation message to continue or set during installation will appear depending on the OS, but please continue as it is.
・ If the computer can not recognize the new hardware due to the security, please go to update the driver from the "Other devices" in the Device Manager.
・ Please perform the installation again from the Device Manager if the driver can't be installed successfully.

5-2.for Reinstallation

・ Please make sure that the communication with all devices that are using this driver is stopped.
・ Please remove the driver from the - "Add or Remove Programs at Control Panel."
（Windows driver Package　（WinUSB）USB Driver from TEXIO）
・ Reinstall Driver by "Setup**.exe".
・ At Device Manager, please make sure that the device you are connected is recognition and working properly.

6. LIST OF FUNCTIONS

6-1. Functions for VC++ use

	Functions	Contents
1	Long USB488_DeviceOpen(int Addr)	Device open
2	int USB488_DeviceClose(int Addr)	Device close
3	int USB488_SetTimeout(int Addr, int time)	Timeout setting
4	int USB488_SetRemote(int Addr, int iMode)	Remote /local setting
5	int USB488_GetRemote(int Addr)	Remote/local reading
6	int USB488_DevClear(int Addr)	Device clear
7	int USB488_GetSTBQuery(int Addr)	Status byte reading
8	int USB488_ClearSTB(int Addr)	Status byte clear
9	int USB488_Send(int Addr, char * sMsg)	String sending
10	int USB488_SendBinary(int Addr ,char * sStream ,int iCount, int iMode)	Binary data sending
11	int USB488_Receive(int Addr,int Count, char *sReceive)	Data receiving

When using API from VC++, refer to definition file for DLL import.
In a sample program, function name is registered without “USB488_”.
When using, initialize and allocate DLL actual function.

6-2. Functions for VC# use

	Functions	Contents
1	Int USB488_DeviceOpen(int Addr)	Device open
2	int USB488_DeviceClose(int Addr)	Device close
3	int USB488_SetTimeout(int Addr, int tTime)	Timeout setting
4	int USB488_SetRemote(int Addr, int iMode)	Remote /local setting
5	int USB488_GetRemote(int Addr)	Remote/local reading
6	int USB488_DevClear(int Addr)	Device clear
7	int USB488_GetSTBQuery(int Addr)	Status byte reading
8	int USB488_ClearSTB(int Addr)	Status byte clear
9	int USB488_Send(int Addr, string strBuf)	String sending
10	int USB488_SendBinary(int addr, [In] byte[ ] pmsg,int count, int nullMode)	Binary data sending
11	int USB488_Receive(int addr, int cnt, StringBuilder pmsg) int USB488_Receive(int addr, int cnt, [In, Out] byte[ ] bytes)	Data receiving

6-3. Functions for the use of VB6.0 and VBA

	Functions	Contents
1	USB488_DeviceOpen (ByVal Addr As Long) As Long	Device open
2	USB488_DeviceClose (ByVal Addr As Long) As Long	Device close
3	USB488_SetTimeout (ByVal Addr As Long, ByVal sec As Long) As Long	Timeout setting
4	USB488_SetRemote (ByVal Addr As Long, ByVal rMode As Long) As Long	Remote/local setting
5	USB488_GetRemote (ByVal Addr As Long) As Long	Remote/local reading
6	USB488_DevClear (ByVal Addr As Long) As Long	Device clear
7	USB488_GetSTBQuery (ByVal Addr As Long) As Long	Status byte reading
8	USB488_ClearSTB (ByVal Addr As Long) As Long	Status byte clear
9	USB488_Send (ByVal Addr As Long, ByVal strbuf As String) As Long	String sending
10	USB488_SendBinary (ByVal Addr As Long, strbuf As Any, ByVal Count As Long, iMode As Long) As Long	Binary data sending
11	USB488_Receive (ByVal Addr As Long, ByVal Count As Long, ByVal strbuf As String) As Long	Data receiving
12	USB488_ReceiveB (ByVal Addr As Long, ByVal Count As Long, strbuf As Any) As Long	Bynary data receiving

When using API functions from VisualBasic6.0 or VBA, use if_60.bas file in source directory.
When using VisualBasic2008/2010, use if_60.vb. In this case, definition of functions is revised according to the change of language specification as per below.

Long → Integer
Any → Byte()

7. DEVICE CONTROL FUNCTIONS

7-1. USB488_DeviceOpen

Function : Open processing is made on the appointed device.
C format : long USB488_DeviceOpen(int Addr)
VB format : USB488_DeviceOpen (ByVal Addr As Long) As Long

Parameter:
Addr : Appointing PC address of the equipment to start communication.
Return value : When device is opened correctly, the value is Windows device handle. When not finding device, the value is “-1”.
Remarks: None

7-2. USB488_DeviceClose

Function: Processing close to the appointed device.
C format: int USB488_DeviceClose(int Addr)
VB format: USB488_DeviceClose (ByVal Addr As Long) As Long

Parameter:
Addr : Appointing PC address for the equipment to terminate communication.
Return value : When function works successfully, “0” is returned. When error is occurred, “-1” is returned.
Remarks : Close device by all means when program terminates. Repeating OPEN and CLOSE continuously with high speed may cause deadlock of communication.

7-3. USB488_SetTimeOut

Function : Setting communication timeout for the appointed device.
C format : int USB488_SetTimeOut(int Addr, int tTime)
VB format : USB488_SetTimeout (ByVal Addr As Long, ByVal sec As Long) As Long

Parameter:
Addr : Appointing PC address for the setting equipment.
Time : Appointing time for timeout（Unit: millisecond）. 0 to 65535.
Timeout is not detected with appointment of 0 msec. Initial value is 1000 msec.
Return value : When function works successfully, “0” is returned, and when error, “-1” is returned.
Remarks : It is required to set longer timeout when using local bus.

7-4. USB488_SetRemote

Function : Setting remote/local for the appointed device.
C format : int USB488_SetRemote(int Addr, int iMode)
VB format : USB488_SetRemote (ByVal Addr As Long, ByVal rMode As Long) As Long

Parameter:
Addr : Appointing PC address for the setting equipment
iMode : Changing local/remote/local lockout status.
0: Local status
1: Remote status（Local key valid）
2: Remote status (All key invalid）
Return value : When function works successfully, “0” is returned, and when error is occurred, “-1” is returned.
Remarks : None

7-5. USB488_GetRemote

Function : Reading remote/local for the appointed device.
C format :int USB488_GetRemote(int Addr)
VB format : USB488_GetRemote (ByVal Addr As Long) As Long

Parameter:
Addr : Appointing PC address of the setting equipment.
Return value : When function works successfully, a positive number is returned, and when failed by error, “-1” is returned.
-1: Device error
0: Local status
1: Remote status（Local key valid）
2: Remote status(Local key invalid)
Remarks : None

7-6. USB488_DevClear

Function : Issuing device clear for the appointed device.
C format : int USB488_DevClear(int Addr)
VB format : USB488_DevClear (ByVal Addr As Long) As Long

Parameter:
Addr : Appointing PC address for the setting equipment.
Return value : When function works successfully, “0” is returned, and when failed by error, “-1” is returned.
Remarks : Device clears the communication buffer. When using local bus, appointment of the local bus is cancelled.

7-7. USB488_GetSTBQuery

Function : Reading status byte for the appointed device.
C format : int USB488_GetSTBQuery(int Addr)
VB format : USB488_GetSTBQuery (ByVal Addr As Long) As Long

Parameter:
Addr : Appointing PC address for the setting equipment.
Return value : When function is successful, status byte is returned, and when failed by error, “-1” is returned. Refer to the unit instruction manual for the contents of status byte.
Remarks : None

7-8. USB488_ClearSTB

Function : Clearing status byte for the appointed device.
C format : int USB488_ClearSTB(int Addr)
VB format : USB488_ClearSTB (ByVal Addr As Long) As Long

Parameter:
Addr : Appointing PC address for the setting equipment.
Return value : When function works successfully, “0” is returned, and when failed by error, “-1” is returned.
Remarks : None

7-9. USB488_Send

Function : Sending string for the appointed device.
C format : int USB488_Send(int Addr , char * sMsg)
VB format : USB488_Send (ByVal Addr As Long, ByVal strbuf As String) As Long

Parameter:
Addr : Appointing PC address for the setting equipment.
sMsg/strbuf : Appointing character to be sent.
Return value : When function is successful, the number of transfer byte is returned, and when failed by error, “-1” is returned.
Remarks : String must be terminated by NULL. Delimiter such as LF is not normally required.
When preparing string with an arrangement, must put “0” in the last place, and confirm the number of transfer byte. String might be UNICODE instead of ASCII depending on the using language. Character code has to be changed from UNICODE to ASCII cord in this case.

7-10. USB488_SendBinary

Function : Sending string for the appointed device.
C format : int USB488_SendBinary(int Addr ,char * sStream , int iCount, int iMode)
VB format : USB488_SendBinary (ByVal Addr As Long, strbuf As Any,ByVal Count As Long, iMode As Long) As Long

Parameter:
Addr : Appointing PC address for the setting equipment.
sStream : Appointing buffer to be sent.
iCount : Appointing byte number to be sent.
iMode : Flag of adding NULL packet for end of transmission.
1: With addition
Return value : When function works successfully, number of transfer byte is returned, and when failed by error, “-1” or “-2” is returned.
Remarks : Control does not return until communication end. When sending large capacity of data, and taking time, transfer them by partial or with another thread for returning control to OS.

7-11. USB488_Receive

Function : Receiving data to the appointed device.
C format : int USB488_Receive(int Addr,int Count, char *sReceive)
VB format : USB488_Receive (ByVal Addr As Long, ByVal Count As Long, ByVal strbuf As String) As Long
: USB488_ReceiveB (ByVal Addr As Long, ByVal Count As Long, strbuf As Any) As Long
Parameter:
Addr : Appointing PC address for the setting equipment.
Count : Appointing byte number for reception request.
sReceive : Appointing buffer to be received.
Functions of USB488_Receive and USB488_ReceiveB are available depending on reception buffer format from VB.
Return value : When function works successfully, transfer byte number is returned, and when failed by error, “-1” or “-2” is returned.
Remarks : Fix and set reception buffer size before calling function not to overflow.
Non-transfer part data of reception buffer is unstable. Cut from the head part by all means by transfer byte number for use.

8. Cautions on use

- Operation speed of the equipment may be slow when continuous setting an reading under high speed clock of PC. When repeating same setting ON/OFF, setting may be seen as omitted because next setting is made before finishing process. Especially internal status and monitor status reading have few hundred msec reading period. Care of that by all means.

- This API is not in conformity with a power saving function for PC, such as suspend. Under the condition of working the power saving function, operation becomes unstable. Set the PC off from the power saving function when using this API.

- When communicating, different GND electric potential between PC and equipment and no GND cause instability of operation. Use it with equipment GND condition using grounded cable by all means. When using USB hub, must install it with the same GND equipment electric potential.

- Communication is unstable under the condition of noise circumstances, such as motor, inverter, welder, etc. are near by. Select location by all means in order not to receive noise effect for PC、controlled equipment, and communication pass, and confirm operation well before use.

- Depending on using programming language, floating point data by binary operation may not be correctly displayed. When comparing values, confirm data format of using language well, and make programming to get the correct value.

- When controlling power supplies from plural number of programs, processes, and threads, verify well that communications are not overlapped.

- Local Bus Connection with plural slave equipments can cause slow communications. Put interval of transmitting lest communications are overlapped.

- If there is “IF_60.dll” in the application directory, operation may become abnormal. Delete DLL from application directly and install API correctly.

- If it is running WindowsUpdate you may not be able to install the USB device driver or API. Please install all again from the end of the process of WindowsUpdate.

TEXIO TECHNOLOGY CORPORATION
7F Towa Fudosan Shin Yokohama Bldg.
2-18-13, Shin Yokohama, Kohoku-ku,Yokohama, Kanagawa, 222-0033 Japan
http://www.texio.co.jp/