

TEXIO

Test and Measurement Solutions

INSTRUCTION MANUAL

For IF-50USB / 50USB

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

Licensing Agreement of the SOFTWARE

1. OUTLINE	1
2. FEATURES	1
3. PRODUCT REQUIREMENT	1
4. INSTALLATION OF API	1
5. INSTALLATION OF USB DEVICE DRIVER	1
5-1.for Installation	1
5-2.for Reinstallation	1
6. LIST OF FUNCTIONS	2
6-1.Function name	2
6-2.Struct Data	3
7-1. LW_DeviceOpen	4
6-2. LW_DeviceClose	4
6-3. Other functions	4
8. Cautions on use	5

1. OUTLINE

- The SOFTWARE consists of a device driver for USB and a module(API) for access, and controls IF-50GP and IF-50USB 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.Function name

Function	Detail
int LW_DeviceOpen (int Class, int BoardAdr, int PcAdr, int SysAdr)	Device Open
int LW_DeviceClose (int DeviceID)	Device Close
int LW_SetTimeOut (int DeviceID, int Time)	Set Timeout
int LW_SetMainInput (int DeviceID, int Flag)	Set Main input Status
int LW_GetMainInput (int DeviceID)	Get Main input Status
int LW_SetInputSelect (int DeviceID, int Ch, int Flag)	Set Selected input Status
int LW_GetInputSelect (int DeviceID, int Ch)	Get Selected input Status
int LW_SetPresetNumber (int DeviceID, int Preset)	Set Preset (Pre1 - Pre4)
int LW_GetPresetNumber (int DeviceID)	Get Preset Status
int LW_SavePreset (int DeviceID, int Src, int Dest)	Copy Preset Data
int LW_GetMonitor (int DeviceID, int Ch, struct MONITOR_DATA *Monitor)	Get Monitor data
int LW_SetMonitorSelect (int DeviceID, int Flag)	Set Monitor Status (A/V,A/W)
int LW_GetMonitorSelect (int DeviceID)	Get Monitor Status
int LW_SetDisplaySelect (int DeviceID, int Ch)	Set Display channel select
int LW_GetDisplaySelect (int DeviceID)	Get selected display channel
int LW_SetLoadMode (int DeviceID, int Preset, int Ch, int Mode, int Ref)	Set Load mode,range,ext ref
int LW_GetLoadMode (int DeviceID, int Preset, int Ch, struct MODE_DATA *ModeData)	Get Load mode
int LW_SetLoadValue (int DeviceID, int Preset, int Ch, double Data)	Set Load value
int LW_GetLoadValue (int DeviceID, int Preset, int Ch, double *Data)	Get Load value
int LW_SetSiemensValue (int DeviceID, int Preset, int Ch, long Data)	Set Siemens Value for CR mode
int LW_GetSiemensValue (int DeviceID, int Preset, int Ch, long *Data)	Get Siemens value for CR mode
int LW_SetCurrentLimit (int DeviceID, int Preset, int Ch, double Data)	Set Current limit
int LW_GetCurrentLimit (int DeviceID, int Preset, int Ch, double *Data)	Get Current limit
int LW_SaveLoadValue (int DeviceID, int Preset, int Ch)	Copy setting to preset
int LW_SaveCurrentLimit (int DeviceID, int Preset, int Ch)	Copy Current limit to preset
int LW_SaveMemoryData (int DeviceID, int object)	Save parameter to memory
int LW_SetSwSelect (int DeviceID, int Ch, int Flag)	Set switching mode
int LW_GetSwSelect (int DeviceID, int Ch)	Get switching mode
int LW_SetSwFreq (int DeviceID, int Freq)	Set switching frequency
int LW_GetSwFreq (int DeviceID, int *Freq)	Get switching frequency
int LW_SetSwDuty (int DeviceID, int Duty)	Set switching duty
int LW_GetSwDuty (int DeviceID, int *Duty)	Get switching duty
int LW_SetSwTime (int DeviceID, double Ta, double Tb)	Set switching time
int LW_GetSwTime (int DeviceID, struct TIME_DATA *Time)	Get switching time
int LW_SetSwMethod (int DeviceID, int Flag)	Set switching mode (freq/time)
int LW_GetSwMethod (int DeviceID)	Get switching mode
int LW_SetDelayFunction (int DeviceID, int Flag)	Set delay mode
int LW_GetDelayFunction (int DeviceID)	Get delay mode
int LW_SetDelayTime (int DeviceID, int Ch, double Time)	Set delay time
int LW_GetDelayTime (int DeviceID, int Ch, double *Time)	Get delay time
int LW_SetTrackFunction (int DeviceID, int Flag)	Set tracking function (on/off)
int LW_GetTrackFunction (int DeviceID)	Get tracking function
int LW_SetTrackDirection (int DeviceID, int ChA, int ChB, int ChC, int ChD)	Set tracking direction
int LW_GetTrackDirection (int DeviceID, struct TRACK_DATA *Track)	Get tracking direction
int LW_SetTrackMode (int DeviceID, int Mode)	Set tracking mode (%/absolute)
int LW_GetTrackMode (int DeviceID)	Get tracking mode
int LW_SetTrackValue (int DeviceID, double Data)	Set tracking value
int LW_GetTrackValue (int DeviceID, double *Data)	Get tracking value
int LW_ResetAlarm (int DeviceID)	Clear alarm
int LW_GetAlarm (int DeviceID, struct ALARM_DATA *Alarm)	Get alarm status
int LW_GetLimit (int DeviceID, struct LIMIT_DATA *Limit)	Get limit value
int LW_SetOperationTimeOut (int DeviceID, int Time)	Set timeout value
int LW_GetOperationTimeOut (int DeviceID)	Get timeout value
int LW_SetLocal (int DeviceID)	Set to local
int LW_SetLocalLockOut (int DeviceID, int Flag)	Set to local lockout
int LW_GetLocalLockOut (int DeviceID)	Get local lockout status
int LW_GetIdNumber (int DeviceID)	Get ID number for LW
int LW_GetChExist (int DeviceID, struct CH_DATA *ChData)	Get channel status for LW

6-2.Struct Data

```
public struct MONITOR_DATA
{
    public double    Current;    // Current Monitor
    public double    Voltage;    // Voltage Monitor
    public double    Wattage;    // Power Monitor
};

public struct MODE_DATA
{
    public int      Mode;        // Load mode
    public int      Reference;   // Load value
};

public struct TIME_DATA
{
    public double    Ta;        // Switching TA
    public double    Tb;        // Switching TB
};

public struct TRACK_DATA
{
    public int      Ch_A;       // Tracking data of CH-A
    public int      Ch_B;       // Tracking data of CH-B
    public int      Ch_C;       // Tracking data of CH-C
    public int      Ch_D;       // Tracking data of CH-D
};

public struct ALARM_DATA
{
    public int      Eta;        // External alarm
    public int      Oha;        // Overheat alarm
    public int      Ova_A;     // Over voltage alarm CH-A
    public int      Ova_B;     // Over voltage alarm CH-B
    public int      Ova_C;     // Over voltage alarm CH-C
    public int      Ova_D;     // Over voltage alarm CH-D
    public int      Oca_A;     // Over current alarm CH-A
    public int      Oca_B;     // Over current alarm CH-B
    public int      Oca_C;     // Over current alarm CH-C
    public int      Oca_D;     // Over current alarm CH-D
    public int      Ora_A;     // Over range alarm CH-A
    public int      Ora_B;     // Over range alarm CH-B
    public int      Ora_C;     // Over range alarm CH-C
    public int      Ora_D;     // Over range alarm CH-D
};

public struct LIMIT_DATA
{
    public int      Current_A;  // Current limit of CH-A
    public int      Current_B;  // Current limit of CH-B
    public int      Current_C;  // Current limit of CH-C
    public int      Current_D;  // Current limit of CH-D
    public int      Power_A;    // Power limit of CH-A
    public int      Power_B;    // Power limit of CH-B
    public int      Power_C;    // Power limit of CH-C
    public int      Power_D;    // Power limit of CH-D
};

public struct CH_DATA
{
    public int      Ch_A;       // CH-A
    public int      Ch_B;       // CH-B
    public int      Ch_C;       // CH-C
    public int      Ch_D;       // CH-D
};
```

7. DEVICE CONTROL FUNCTIONS

7-1. LW_DeviceOpen

Function: Device Open。

format: int LW_DeviceOpen(int Class, int BoardAdr, int PcAdr, int SysAdr)

parameter:

- Class
Device Select GP-IB or USB.
 - 1 - GPIB
 - 2 – USB
- BoardAdrs
GPIB Bourd ID for National Instrument. 0~3
USB : 0 (fixed)。
- PcAdrs
PC Address for LW (0~31)
- SysAdrs
SYStem Address for LW (1~32)

Return:

Value: Deice ID

- 1 – Device Error
- 2 – Timeout Error
- 3 – Parameter Error

7-2. LW_DeviceClose

Function :Close Device

Format : int LW_DeviceClose(int DeviceID)

Parameter:

- DeviceId
Opened Device ID.

Return:

- 0 – Complete
- 1 – Device Error
- 2 – Timeout Error

7-3. Other functions

Please refer sample application.

8. Cautions on use

- Operation speed of the equipment may be slow when continuously setting a 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_50.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/>
