How to download and compile a custom C program?

[Download FAQ-021 demo]

In this section, we use the C program “LP2648M_SM_demo” developed on the LP-2648M and the SmarTTY tool to upload, compile, and run the program. We also explain how to read/write data via shared memory and interact with variables in a Win-GRAF project.

SmarTTY is an SSH client connection tool. The following operations are performed in a Windows environment. Use SmarTTY to connect to the LP-2648M to upload and compile the program.

Please download and install SmarTTY first: https://sysprogs.com/SmarTTY/download/

LP2648M_SM_demo Setup and Notes

1. Connect to the LP-2648M via SmarTTY

a. Click "New SSH Connection".

b. Enter the following information:

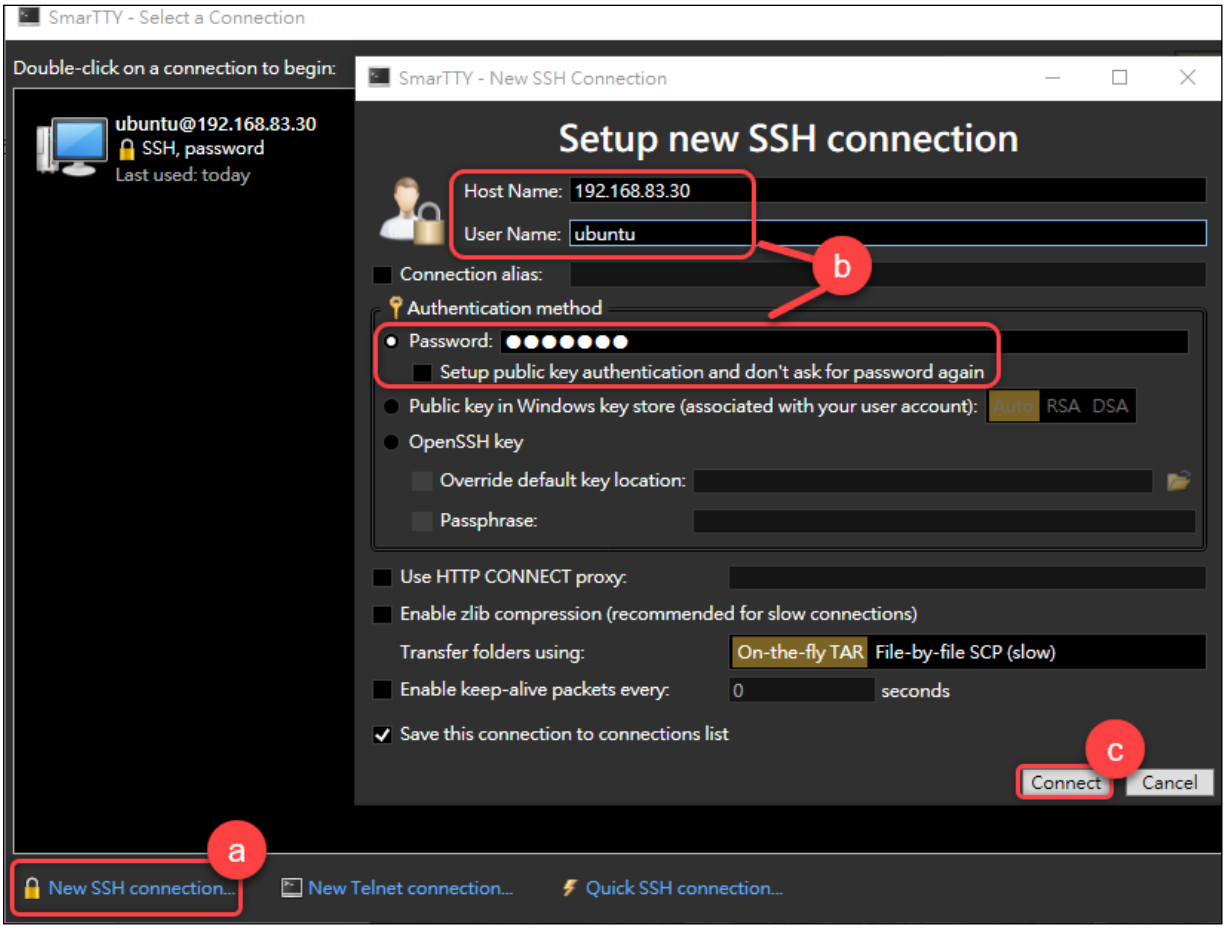
Host Name: 192.168.83.30

User Name: ubuntu

Password: temppwd

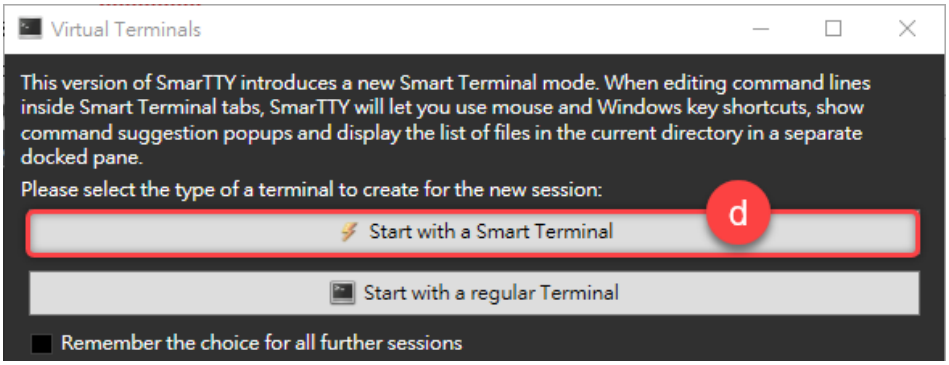
Uncheck”Setup public key”

c. Click "Connect".

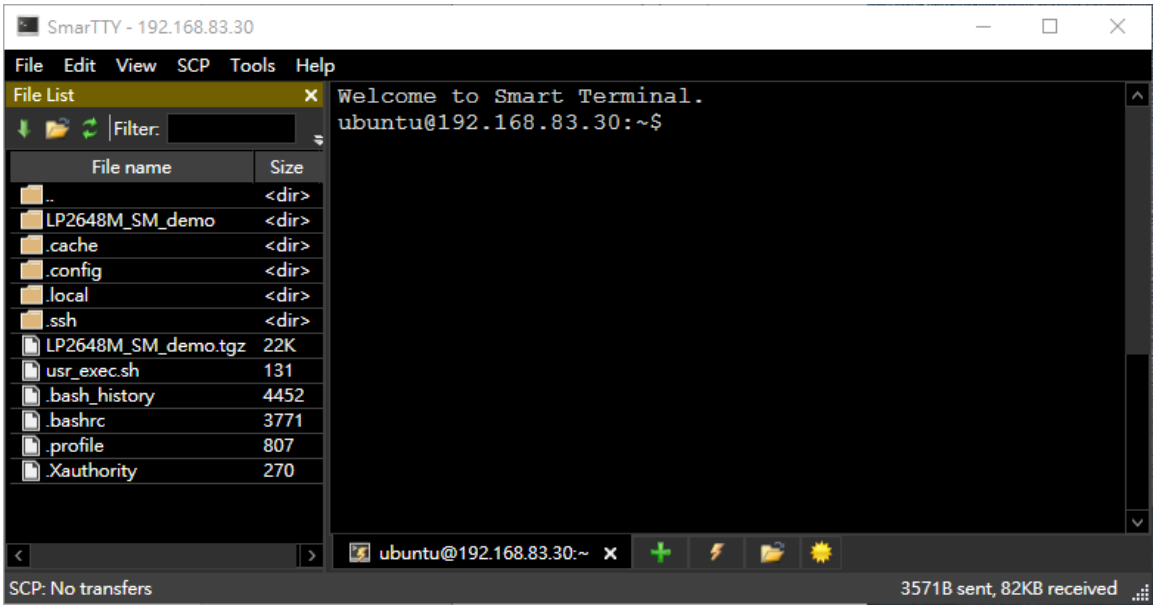


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

d. Click "Start with a Smart Terminal".



A successful connection will show the following screen.

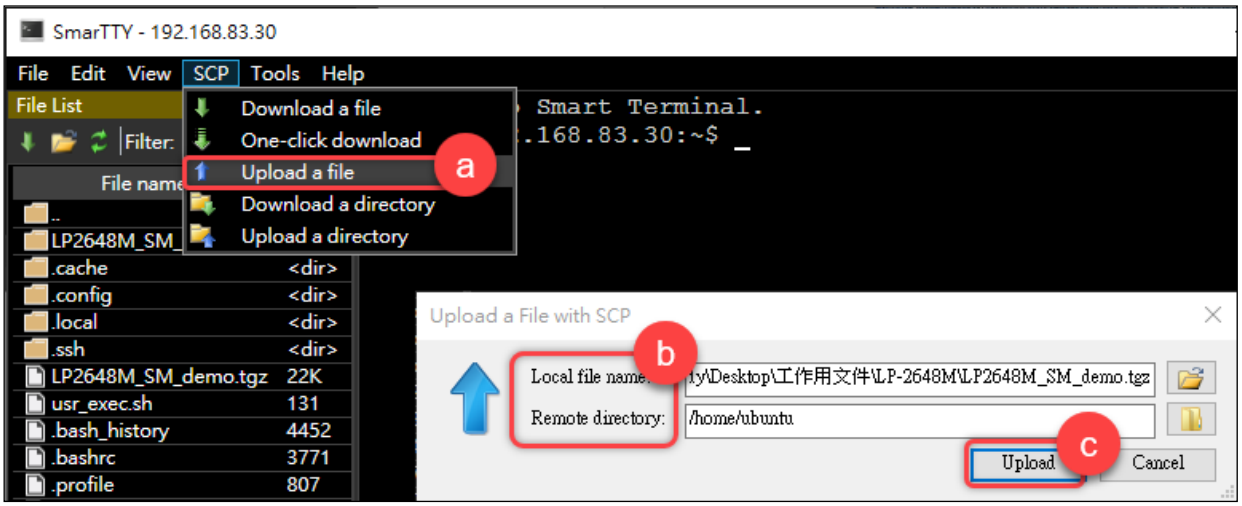


2. Upload the program

a. Click SCP -> Upload a file

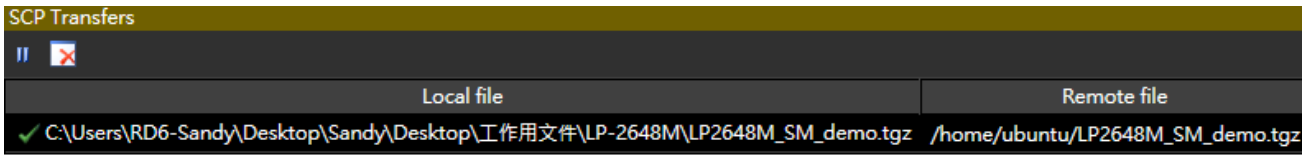
b. Select the file and the destination path

c. Click "Upload".



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

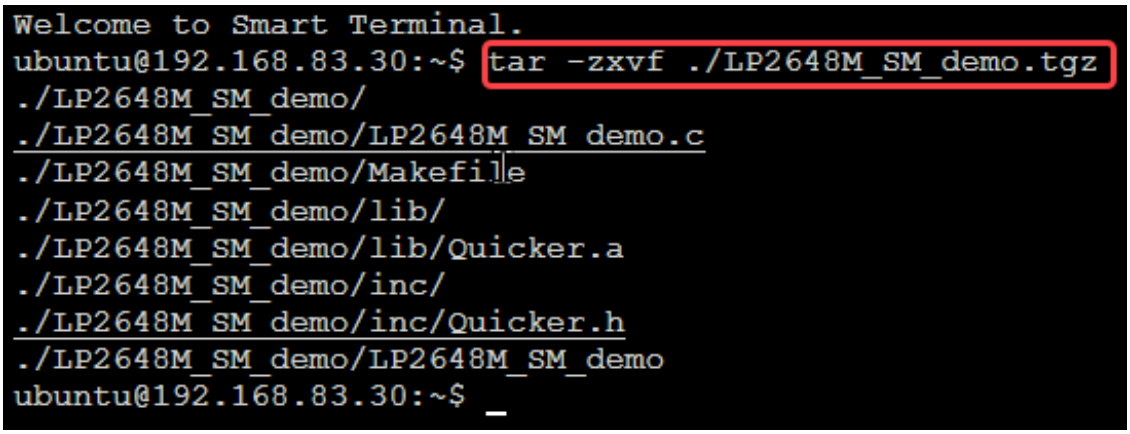
If the file is uploaded successfully, the following figure will appear.



√ means uploaded successfully

3. Unzipped file name: LP2648M_SM_demo.tgz

Use the command **tar -zxvf ./LP2648M_SM_demo.tgz** to unzip

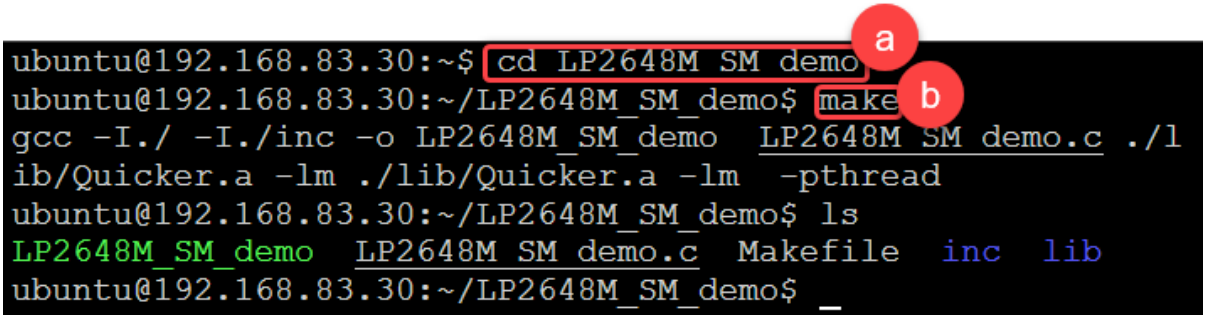


4. Compile

a. Use the command **cd LP2648M_SM_demo** to switch to the LP2648M_SM_demo folder

b. Use the command **make** to compile/build the program

(If running **make** and returns **Nothing to be done for 'all'**, it means the program has already been compiled/built and there are no file changes, so you do not need to rebuild. Proceed to the next step.)

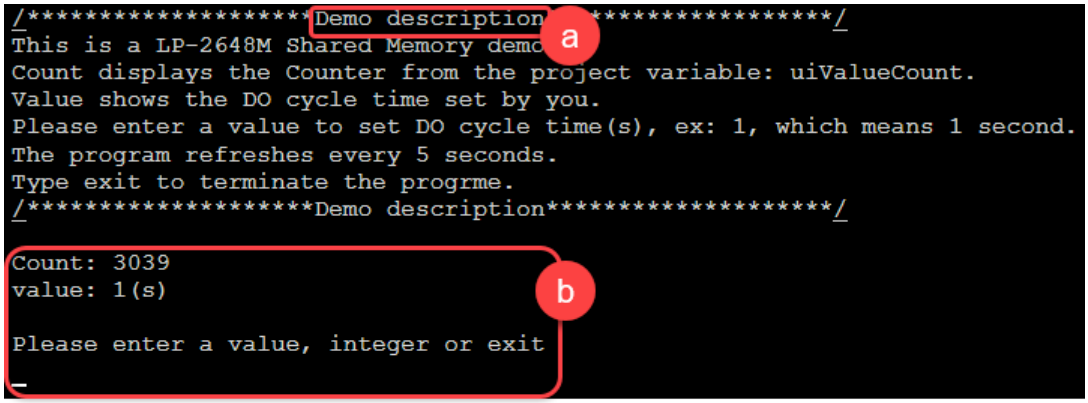


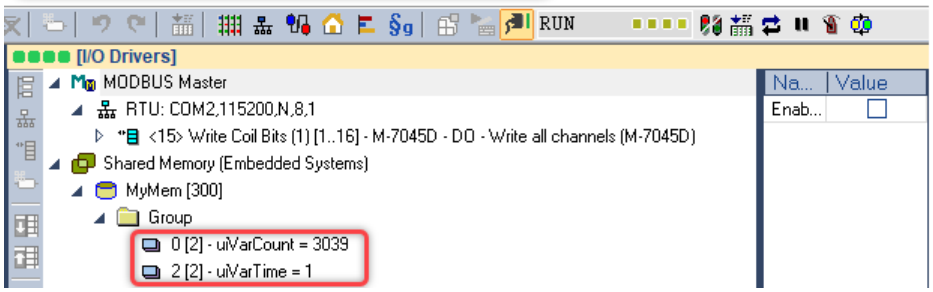
5. Run the program

※Note: Please download WG_LP2648M_SM_demo from Win-GRAF to the LP-2648M before performing the steps below.

Use the command **./LP2648M_SM_demo** to run the program

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

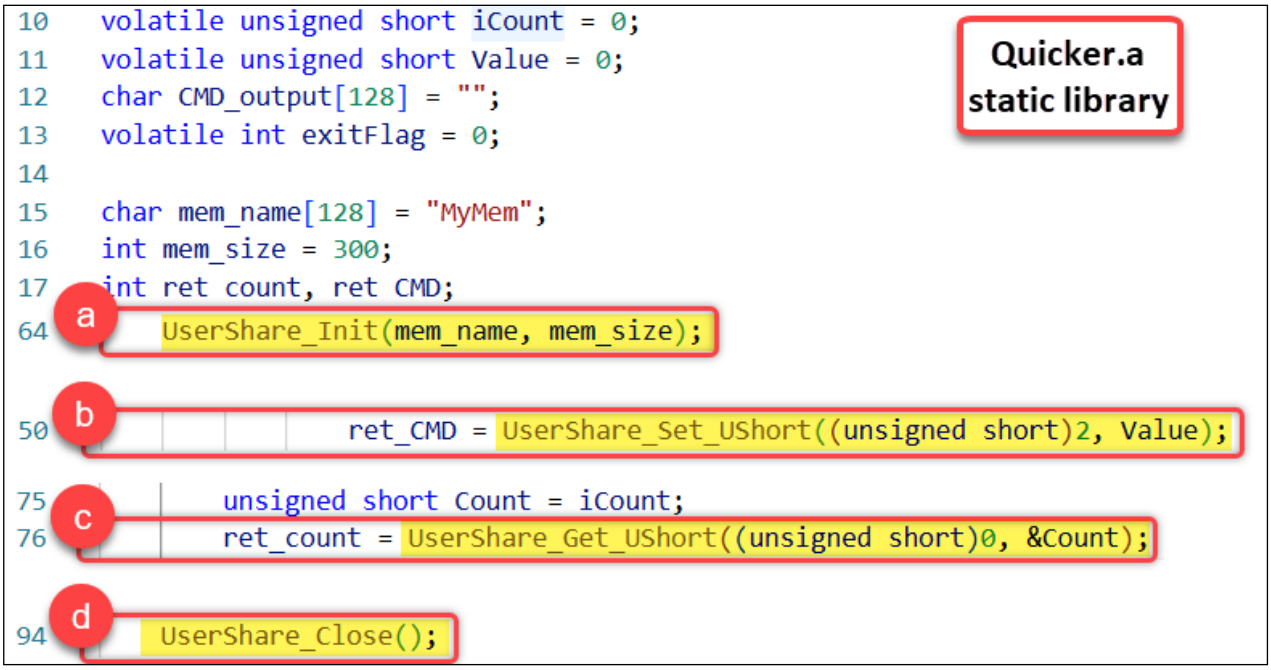




a. Demo description
b. Count: Displays the value of the Win-GRAF demo variable uiVarCount
(In this demo program, the value on the screen is automatically refreshed every 5 seconds.)
value: Displays the value written to uiVarTime.
→Please enter a value, integer or exit
(Enter the value to write to uiVarTime here. Enter **exit** to end the program.)

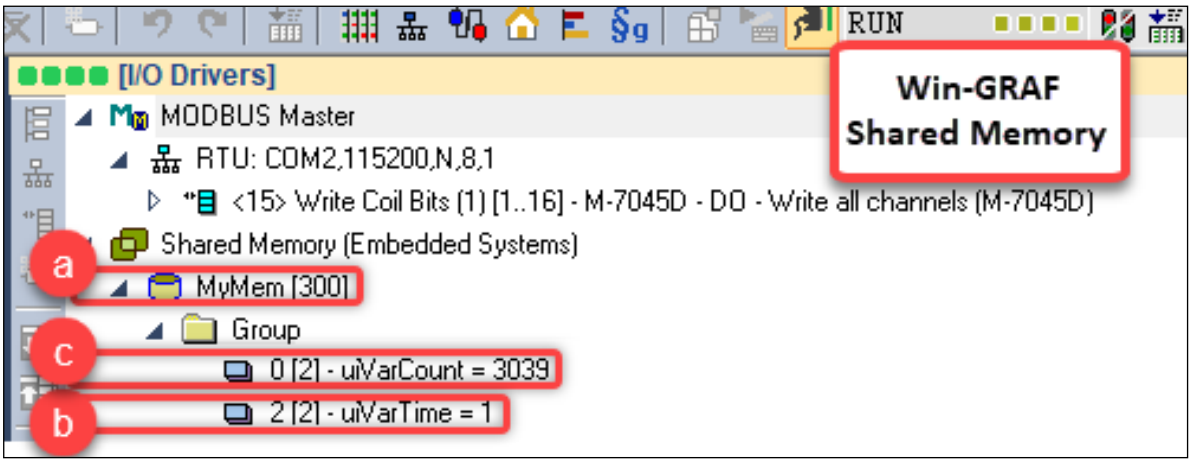
6. Program description

LP2648M_SM_demo uses the **Quicker.a** static library to implement shared memory



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

The **Quicker.a static library** corresponds to Win-GRAF's Shared Memo feature, as shown in the figure below:



The following APIs are used in the demo:

a. **void UserShare_Init(char *mmap_name, int mmap_size)**
Connect to the shared memory based on the Win-GRAF Shared Memory name: MyMem, size: 300 bytes.

b. **unsigned char UserShare_Set_UShort(unsigned short iUserAddress, unsigned short iStatus)**
Write an unsigned short value to the Win-GARF demo bound variable uiVarTime. This variable is located at offset 2 in the "MyMem" shared memory, so call it as **UserShare_Set_UShort((unsigned short)2, value)**

c. **unsigned char UserShare_Get_UShort(unsigned short iUserAddress, unsigned short *iStatus)**
Read an unsigned short value from the Win-GARF demo bound variable uiVarCount. This variable is located at offset 0 in the "MyMem" shared memory, so call it as **UserShare_Get_UShort((unsigned short)0, &Count)**

d. **void UserShare_Close(void)**
End the use of shared memory.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Quicker.a API Description

"Quicker.a" provides various functions to operate Win-GRAF variable data, including:

- ✧ Connect/close Shared Memory
- ✧ Read/write Boolean
- ✧ Read/write 8-bit, 16-bit, 32-bit, or 64-bit Integer
- ✧ Read/write 32-bit or 64-bit REAL
- ✧ Read/write up to 256 characters

- **Connect/close Shared Memory**

a. void UserShare_Init(char *mmap_name, int mmap_size);
mmap_name: The shared memory name configured in Win-GRAF Shared Memory.
mmap_size: The shared memory size configured in Win-GRAF Shared Memory.

b. void UserShare_Close(void);
Call this function before the program ends.

- **Shared Memory API Read/Write Mapping Table**

API I/O Type	Win-GRAF Type	Data Size	API Name
Set	BOOL	8 bits	unsigned char UserShare_Set_Coil (unsigned short iUserAddress, unsigned char iStatus);
	INT	16 bits	unsigned char UserShare_Set_Short (unsigned short iUserAddress, short iStatus);
	UINT	16 bits	unsigned char UserShare_Set_UShort (unsigned short iUserAddress, unsigned short iStatus);
	LINT	64 bits	unsigned char UserShare_Set_Int64 (unsigned short iUserAddress, long long iStatus);
	ULINT	64 bits	unsigned char UserShare_Set_UInt64 (unsigned short iUserAddress, unsigned long long iStatus);
	DINT	32 bits	unsigned char UserShare_Set_Long (unsigned short iUserAddress, long iStatus);
	UDINT	32 bits	unsigned char UserShare_Set_ULong (unsigned short iUserAddress, unsigned long iStatus);
	SINT	8 bits	unsigned char UserShare_Set_Char (unsigned short iUserAddress, char iStatus);
	USINT	8 bits	unsigned char UserShare_Set_UChar (unsigned short iUserAddress, unsigned char iStatus);
	REAL	32 bits	unsigned char UserShare_Set_Float (unsigned short iUserAddress, float iStatus);

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

API I/O Type	Win-GRAF Type	Data Size	API Name
Set	LREAL	64 bits	unsigned char UserShare_Set_Double (unsigned short iUserAddress, double iStatus);
	STRING	256 bits or customizable	int UserSetReg_Str (unsigned short iUserAddress, char iStatus, unsigned short len);
Get	BOOL	8 bits	unsigned char UserShare_Get_Coil (unsigned short iUserAddress, unsigned char *iStatus);
	INT	16 bits	unsigned char UserShare_Get_Short (unsigned short iUserAddress, short *iStatus);
	UINT	16 bits	unsigned char UserShare_Get_UShort (unsigned short iUserAddress, unsigned short *iStatus);
	LINT	64 bits	unsigned char UserShare_Get_Int64 (unsigned short iUserAddress, long long *iStatus);
	ULINT	64 bits	unsigned char UserShare_Get_UInt64 (unsigned short iUserAddress, unsigned long long *iStatus);
	DINT	32 bits	unsigned char UserShare_Get_Long (unsigned short iUserAddress, long *iStatus);
	UDINT	32 bits	unsigned char UserShare_Get_ULong (unsigned short iUserAddress, unsigned long *iStatus);
	SINT	8 bits	unsigned char UserShare_Get_Char (unsigned short iUserAddress, char *iStatus);
	USINT	8 bits	unsigned char UserShare_Get_UChar (unsigned short iUserAddress, unsigned char *iStatus);
	REAL	32 bits	unsigned char UserShare_Get_Float (unsigned short iUserAddress, float *iStatus);
	LREAL	64 bits	unsigned char UserShare_Get_Double (unsigned short iUserAddress, double *iStatus);
	STRING	256 bits or customizable	int UserGetReg_Str (unsigned short iUserAddress, char *iStatus, unsigned short len);

❖ **In the APIs above, iUserAddress represents the Win-GRAF offset address.**

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000