

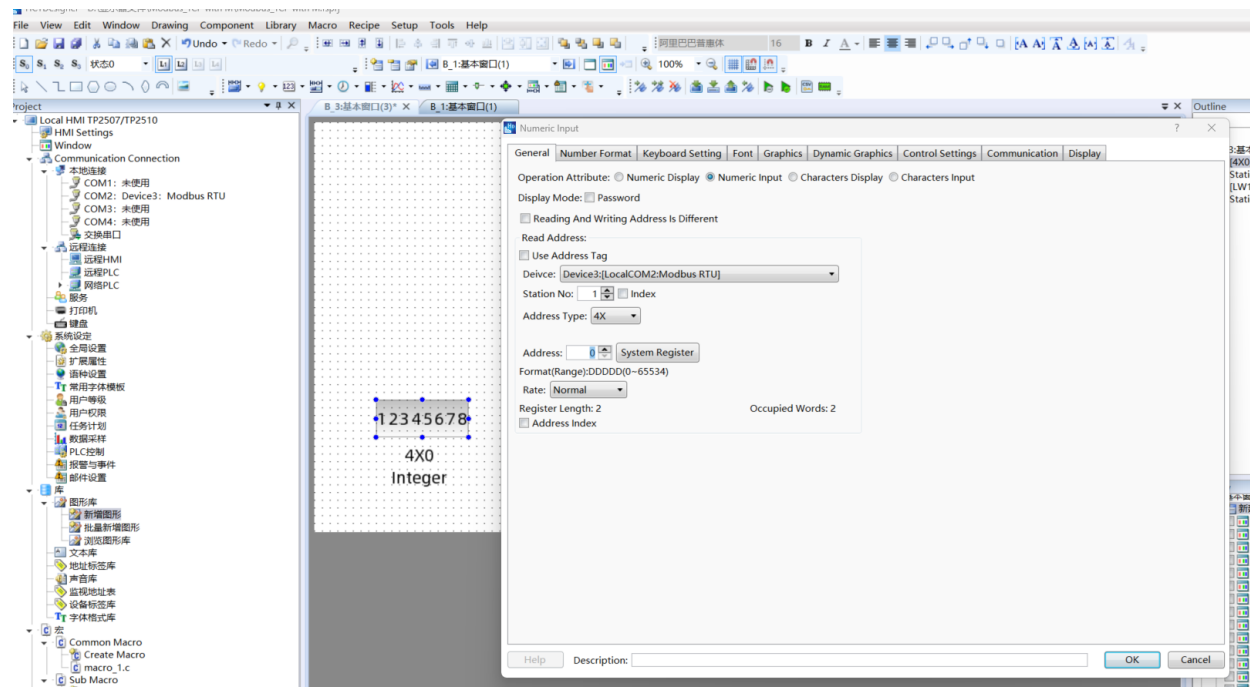
TP2000系列触摸屏,整形变量转换为ASCII

TP2000 series touch screen, convert Integer variables into ASCII

1.创建元件/Creat components

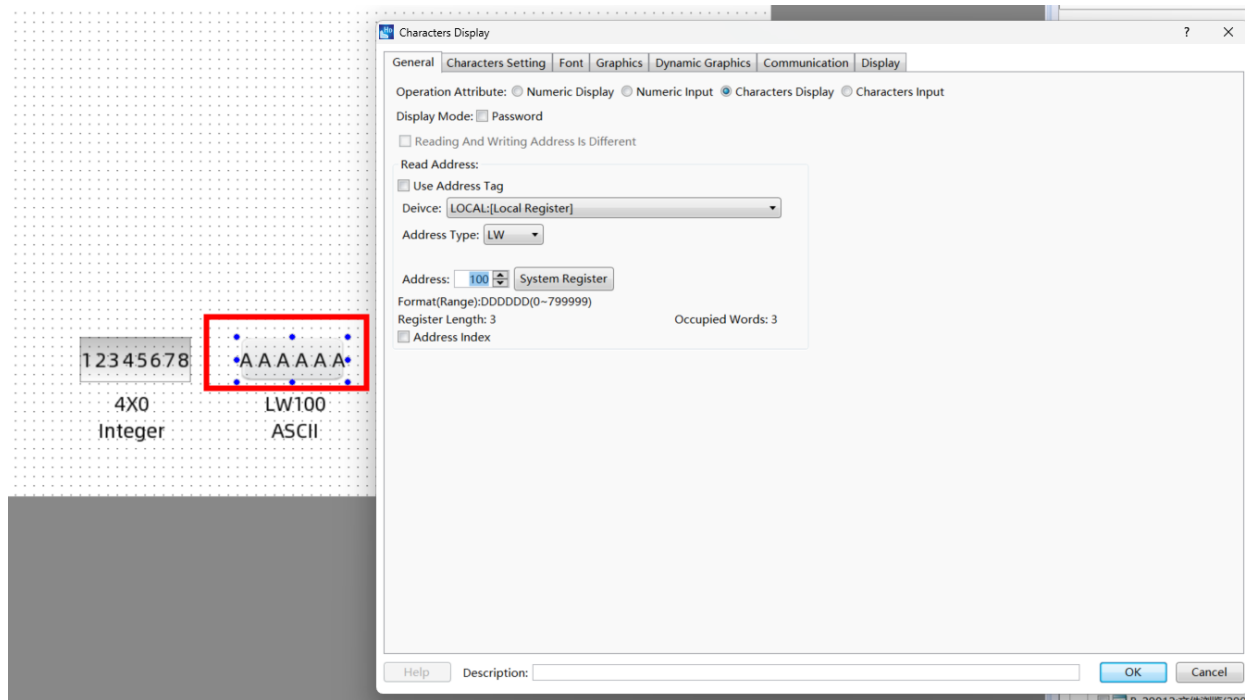
1.整数输入

1.Integer input



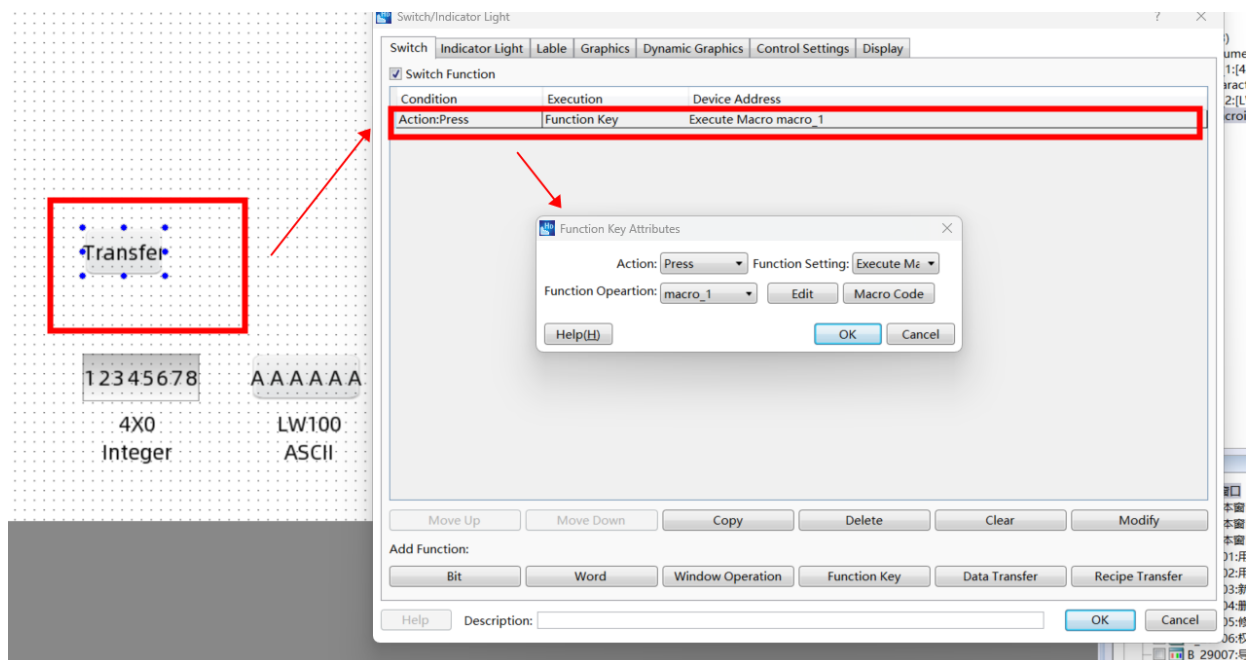
2.ASCII码显示

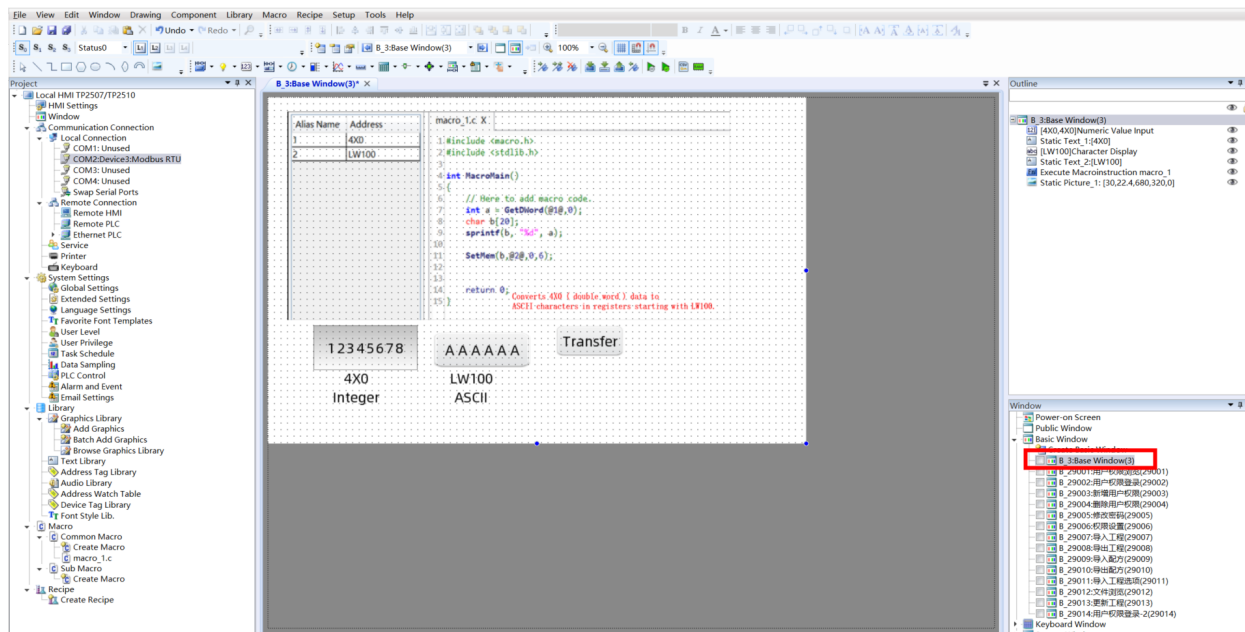
2.ASCII Display



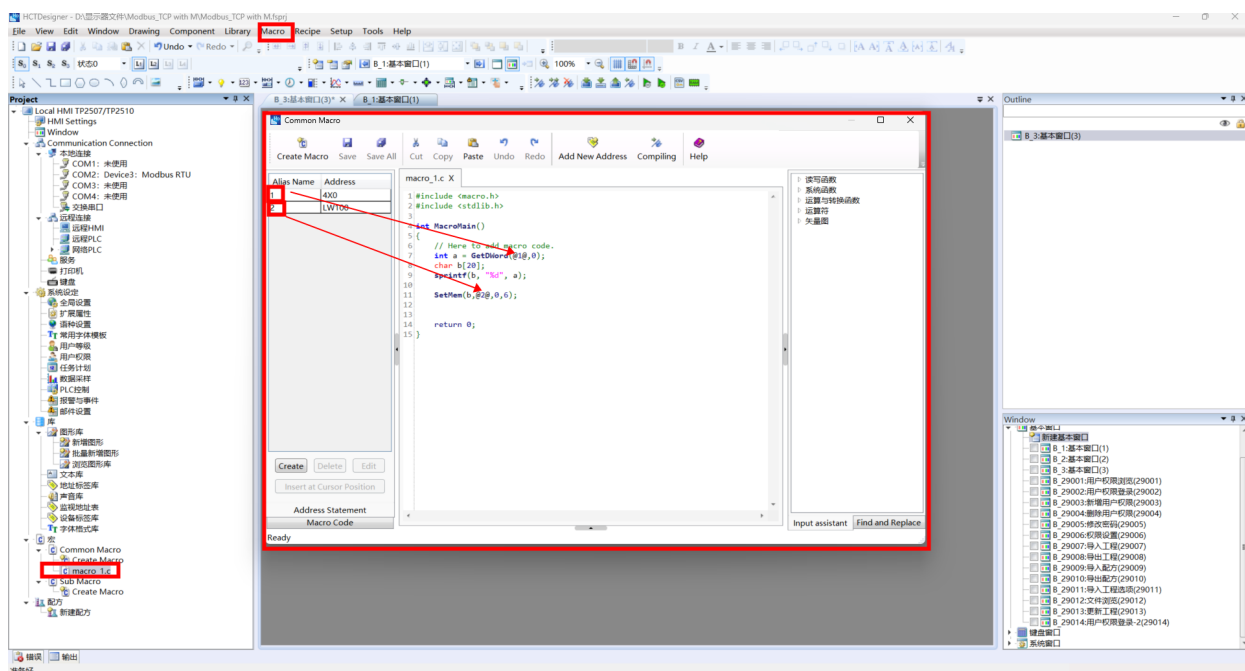
3.宏的功能键设定

3.Function key setting





2.创建宏指令，准备程序/Creat a Macro, and write program



图中程序为将整形变量转换为ASCII

程序 (Program) :

```
#include <macro.h>
#include <stdlib.h>

int MacroMain()
{
    // Here to add macro code.
    int a = GetDWord(@1@,0);
```

```
char b[20];
sprintf(b, "%d", a);

SetMem(b,@2@,0,6);

return 0;
}
```

C语言里面有两个函数atoi (Ascii转整形) 和atof (Ascii转浮点型) , 使用这两个函数即可把Ascii转换成整形和浮点型数据。这两个函数的头文件是<stdlib.h>。

函数格式:

Int a=atoi(str);

float b=atof(str);

注意: 待转换字符串必须是0-9或者".", 不能有其他的Ascii值, 不然会报错

There are two functions in C language, atoi (Ascii trans-shaping) and atof (Ascii trans-floating point type). Using these two functions, Ascii can be converted.

Shaping and floating-point data. The header files of these two functions are.

Function format :

Int a = atoi (str);

float b = atof (str);

Note : The string to be converted must be 0-9 or ' . ', and there must be no other Ascii values, otherwise an error will be reported.

GetDWord: 读取DWORD地址/Get address of DWORD

Common Macro

Create Macro Save Save All Cut Copy Paste Undo Redo Add New Address Compiling Help

Alias Name	Address
1	4X0
2	LW100

macro_1.c * X

```
1 #include <macro.h>
2 #include <stdlib.h>
3
4 int MacroMain()
5 {
6     // Here to add macro code.
7     int a = GetDWord(@1@,0);
8     char b[20];
9     sprintf(b, "%d", a);
10
11     SetMem(b,@2@,0,6);
12 }
```

Use direction

unsigned int GetDWord(@地址别名@.地址偏移):

从指定寄存器地址读取一个双字。
特别注意: 如果读取的地址为PLC的外部变量, 请确保使用GetError函数确保本函数执行成功, 否则可能会导致 @地址别名@: 选一个双字地址类型
地址偏移: 正整数 (unsigned int), 读取地址=@地址别名@.声明的地址+地址偏移
返回值: unsigned int型, 读取的双字地址的值。如果是字类型, 读取连续的两个字, 小端序, 低位在前

错误信息:
使用GetError函数获得错误码。
int error=GetError();

使用示例:
unsigned int speed=GetDWord(@速度@.3);
@速度@地址设定为LW0, 则本例表示从LW3和LW4读取双字, 返回到unsigned int型变量speed
//如果使用其它数据类型, 可以使用强制转换
int speed = (int) GetDWord(@速度@.3);

Read Word Register

Address Alias 1

Address Offset

Parameter Type: Constant

Constant Value 0

Data Type: unsigned int

Return Value

Variable Name

Data Type: unsigned int

Code Preview

GetDWord(@1@,0);

Input assistant Information List

Ready

读写函数

- 读取位地址: GetBit
- 读取字地址: GetWord
- 读取双字地址: GetDWord
- 读取浮点数: GetFloat
- 读取连续地址: GetMem
- 按照数据类型读取连续地址: GetMemEx
- 设定位地址: SetBit
- 设定字地址: SetWord
- 设定双字地址: SetDWord
- 设定浮点数: SetFloat
- 设定连续地址: SetMem
- 按照数据类型设定连续地址: SetMemEx

系统函数

- 运算与转换函数
- 运算符
- 矢量图

SetMem: 设定连续地址/Set a continuous address

Common Macro

Create Macro Save Save All Cut Copy Paste Undo Redo Add New Address Compiling Help

Alias Name	Address
1	4X0
2	LW100

macro_1.c * X

```
1 #include <macro.h>
2 #include <stdlib.h>
3
4 int MacroMain()
5 {
6     // Here to add macro code.
7     int a = GetDWord(@1@,0);
8     char b[20];
9     sprintf(b, "%d", a);
10
11     SetMem(b,@2@,0,6);
12 }
```

Use direction

int SetMem(数组指针.@地址别名@.地址偏移.字节数):

将数组数据设定到指定起始地址的连续寄存器。
数组指针: 指针类型, 指向预先声明的数组。
特别注意: 如果读取的地址为PLC的外部变量, 请确保使用GetError函数确保本函数执行成功, 否则可能会导致 @地址别名@: 选一个寄存器类型的起始地址, 可以是位类型也可以是字类型
地址偏移: 正整数 (unsigned int), 读取的起始地址=@地址别名@.声明的地址+地址偏移
字节数: 正整数 (unsigned int), 要读取的字节数。特别注意需要根据数据类型换算。
字节数近似是sizeof(数组声明的数据类型)*要读取的数组个数。字节数最大值20480
返回值: int型, 返回值 1-成功, 0-读取失败, (-4)-调用失败

使用示例:
unsigned short data[10];
int error = SetMem(data,@数组数据@.2,10*sizeof(unsigned short));
@数组数据@地址设定为LW0, 则本例表示将数组data中的数据复制到LW2起始的10个寄存器中。
char data[5];
int error = SetMem(data,@数组数据@.2,5);
@数组数据@地址设定为LW0, 则本例表示将data中5个字节的数据复制到从LW2起的3个字中(每个字包含两个ch

Array Pointer

Variable Name b

Data Type: void *

Write Address

Address Alias 2

Address Offset

Parameter Type: Constant

Constant Value 0

Data Type: unsigned int

Writing Byte Number

Parameter Type: Constant

Constant Value 6

Data Type: unsigned int

Return Value

Variable Name

Code Preview

SetMem(b,@2@,0,6);

Input assistant Information List

Ready

读写函数

- 读取位地址: GetBit
- 读取字地址: GetWord
- 读取双字地址: GetDWord
- 读取浮点数: GetFloat
- 读取连续地址: GetMem
- 按照数据类型读取连续地址: GetMemEx
- 设定位地址: SetBit
- 设定字地址: SetWord
- 设定双字地址: SetDWord
- 设定浮点数: SetFloat
- 设定连续地址: SetMem
- 按照数据类型设定连续地址: SetMemEx

系统函数

- 运算与转换函数
- 运算符
- 矢量图