

作者： 陈颂文

日期： 2016/7/7

版本： V3.0

E\_mail: s.chen@beckhoff.com.cn

**BECKHOFF** New Automation Technology

上海市江场三路市北工业园区

163 号 5 楼 (200436)

TEL: 020-38010300

FAX: 020-38010303

## 基于 TC3 的 PLC HMI 配方功能

---

### 概 述

技术说明文档模板，为了使测试信息更加可靠，需要详细描述所用的硬件和软件版本，包含但不限于以下内容。

### 文档中包含的文件

| 文件名称                 | 文件说明         |
|----------------------|--------------|
| 《TC3_PLC_HMI_recipe》 | 测试所用的 plc 程序 |
|                      |              |
|                      |              |

### 备 注

关键字： TC3  
HMI  
配方

### 免责声明

我们已对本文档描述的内容做测试。但是差错在所难免，无法保证绝对正确并完全满足您的使用需求。本文档的内容可能随时更新，也欢迎您提出改进建议。

*文档内容可能随时更新  
如有改动，恕不事先通知*

## 基于 TC3 的 PLC HMI 配方功能 V3.0

### 一、测试设备

#### 1. 硬件设备:

工程师自用笔记本电脑。32 位 Win7 操作系统测试 OK。

C69 工控机。64 位 Win7 操作系统测试 OK。

#### 2. 软件版本:

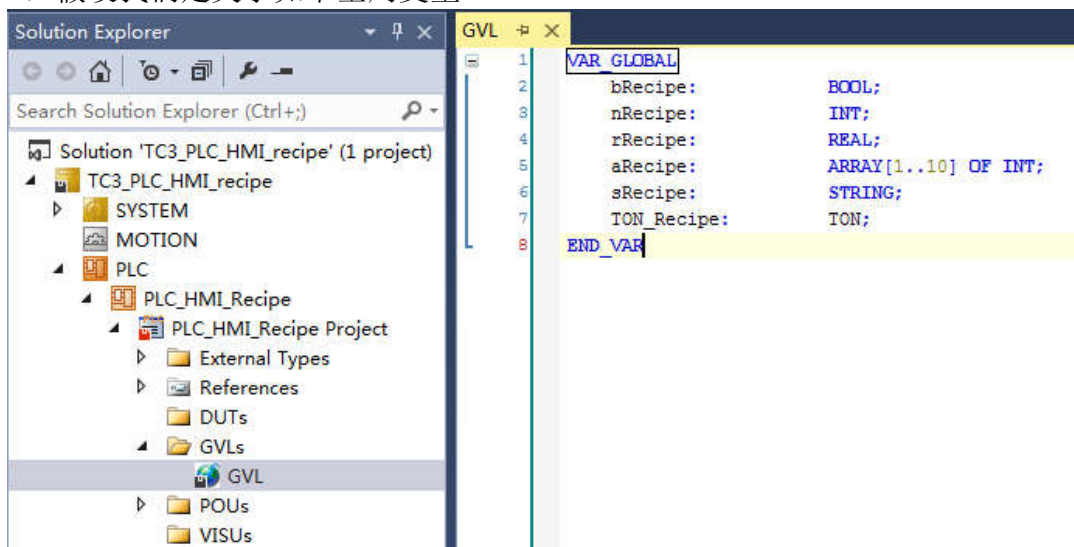


TwinCAT 版本号为 3.1.4020.32

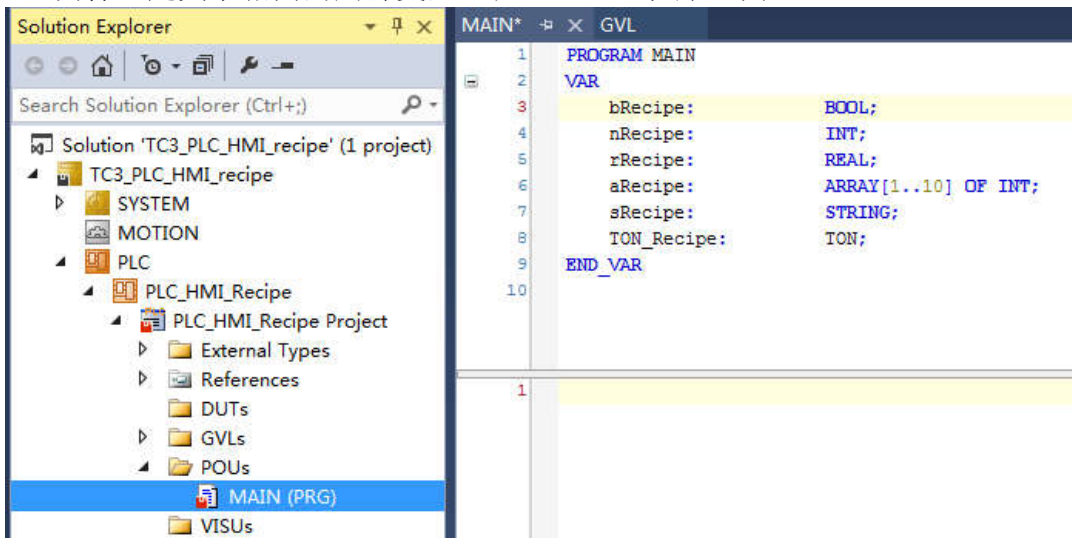
### 二、正文:

#### 一、创建变量

1) 假设我们定义了如下全局变量:

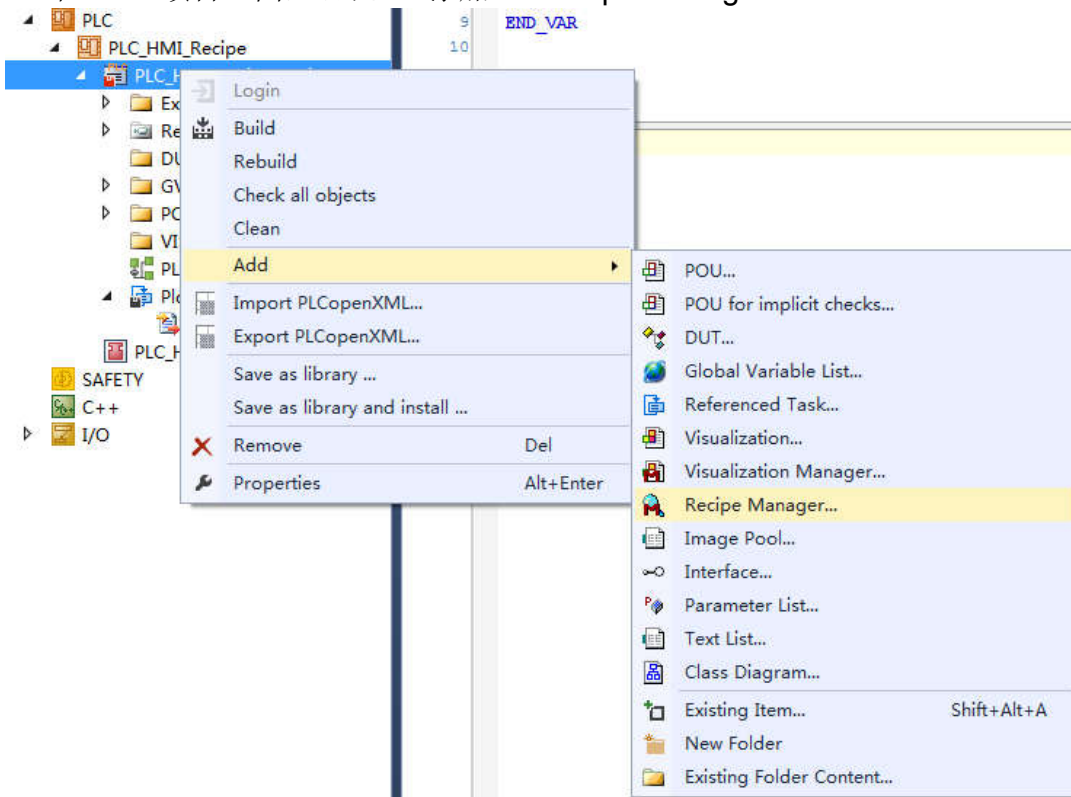


2) 同样，定义了相同的局部变量，在 MAIN 主程序里面：



## 二、创建 Recipe Manager

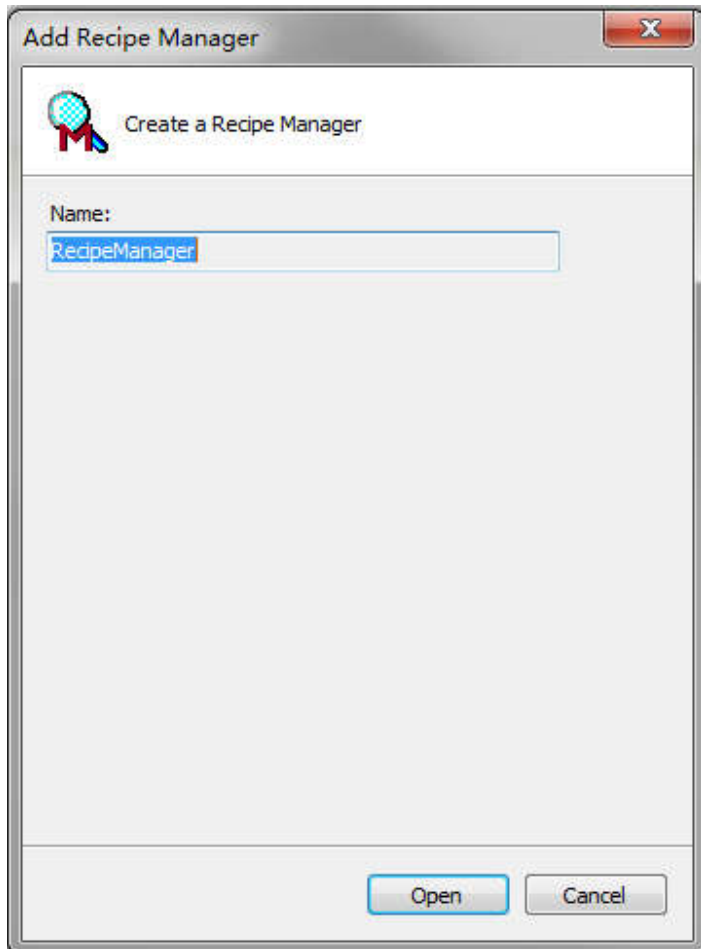
3) 在 PLC 项目上面，右键—>添加—>Recipe Manager



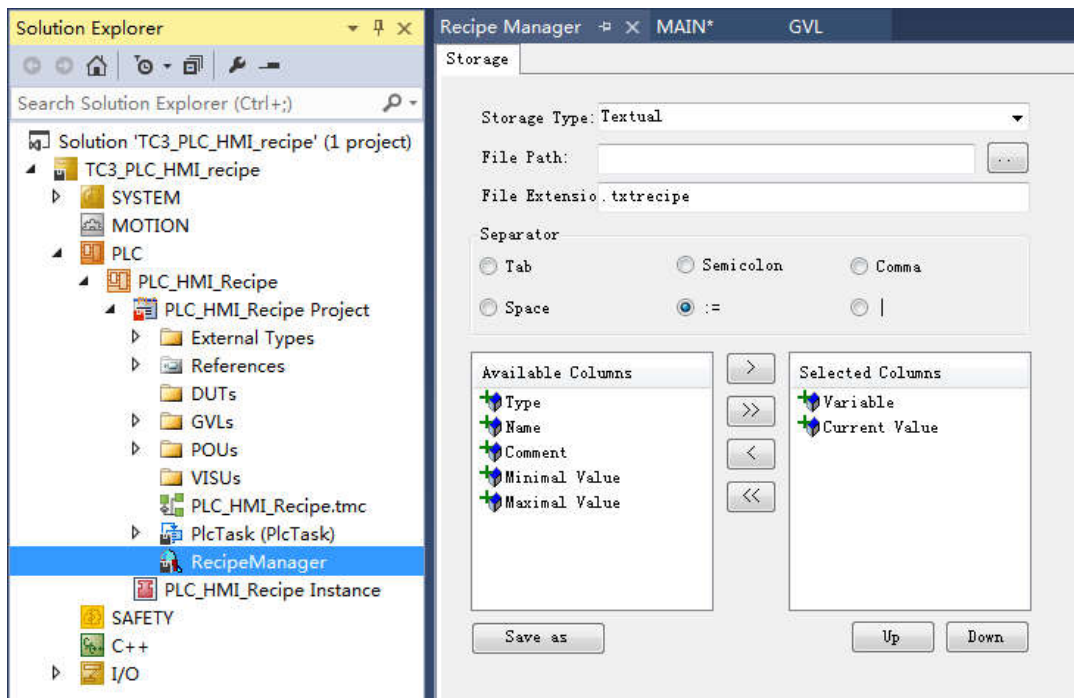
# BECKHOFF

---

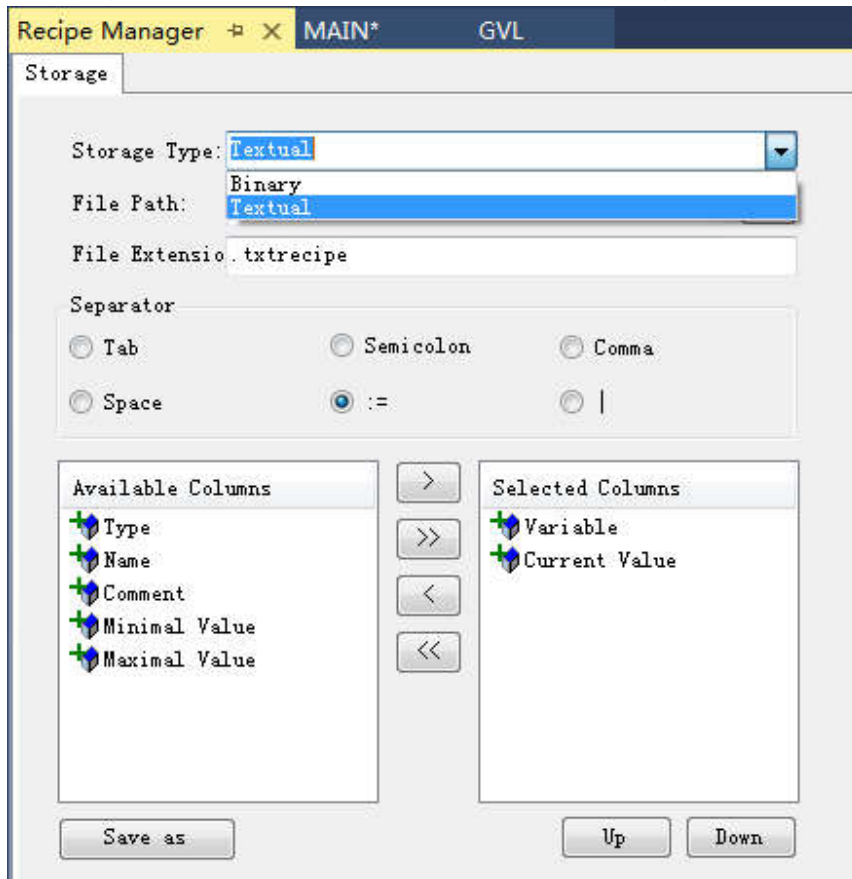
4) 弹出对话框，可以输入名字。本例有默认名。然后点击 ‘Open’ 按钮。



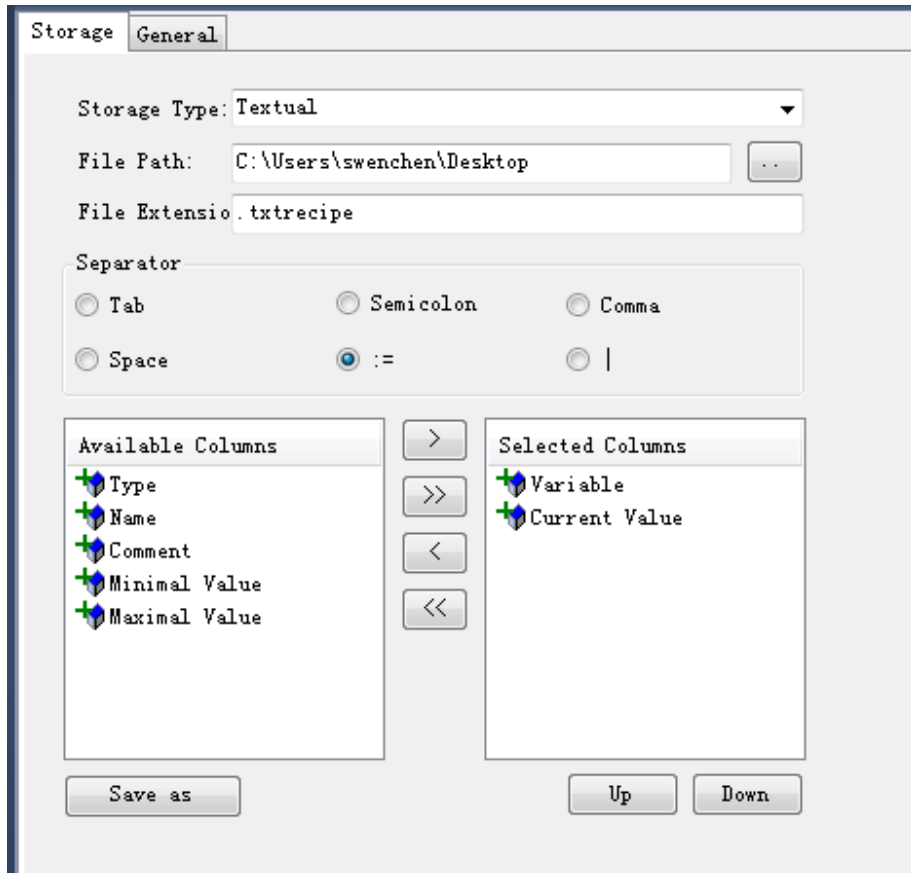
5) 创建好之后，



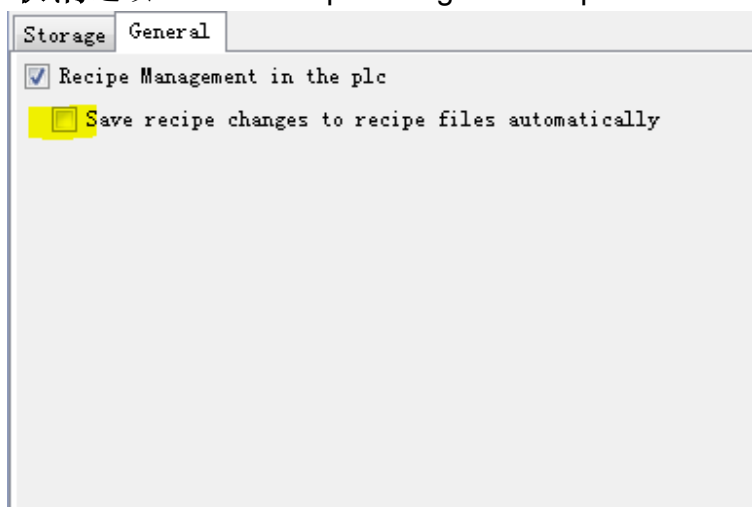
6) 有两种方式保存配方文件的格式。使用默认的 **txt** 格式，方便以后对配方文件进行读取、修改。如果是 **bin** 格式，则不能用其他软件读取、修改保存的配方文件。



注意事项：FilePath 可以不用设置！

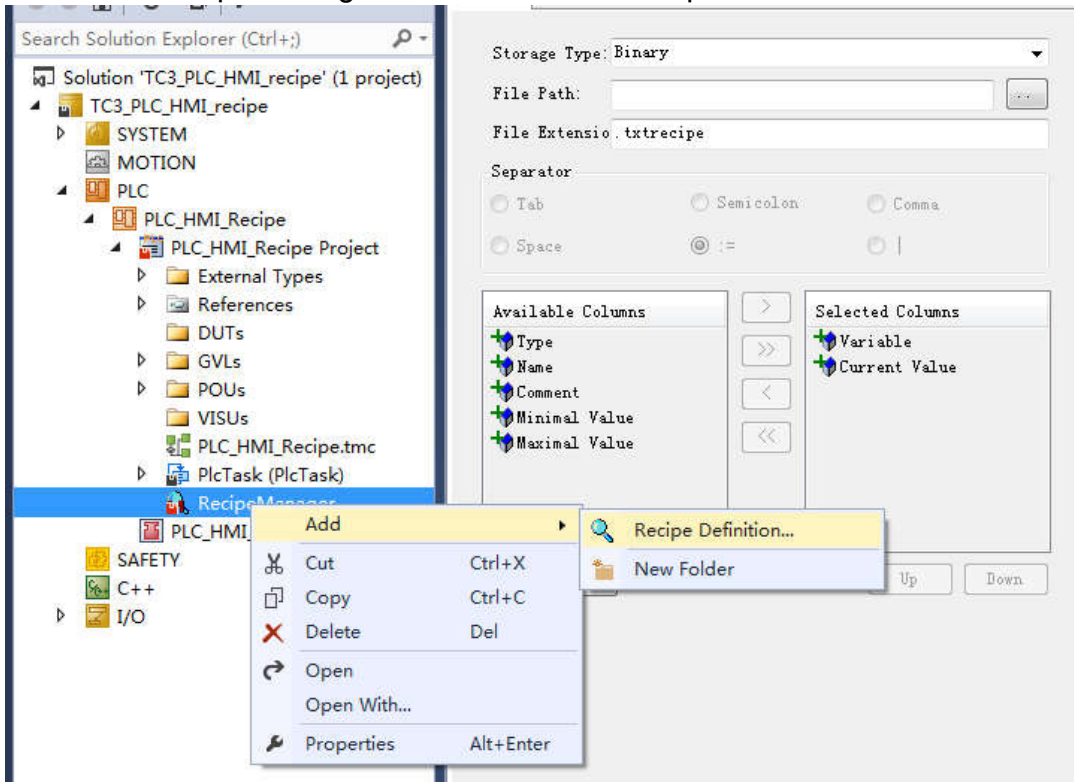


取消选项 “Save Recipe changes to recipe files automatically”

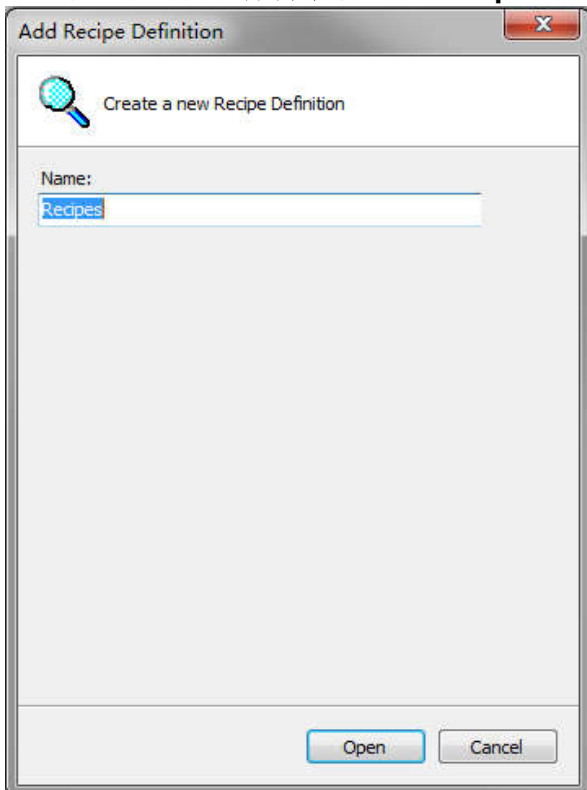


注：请查看附录的调试笔记。

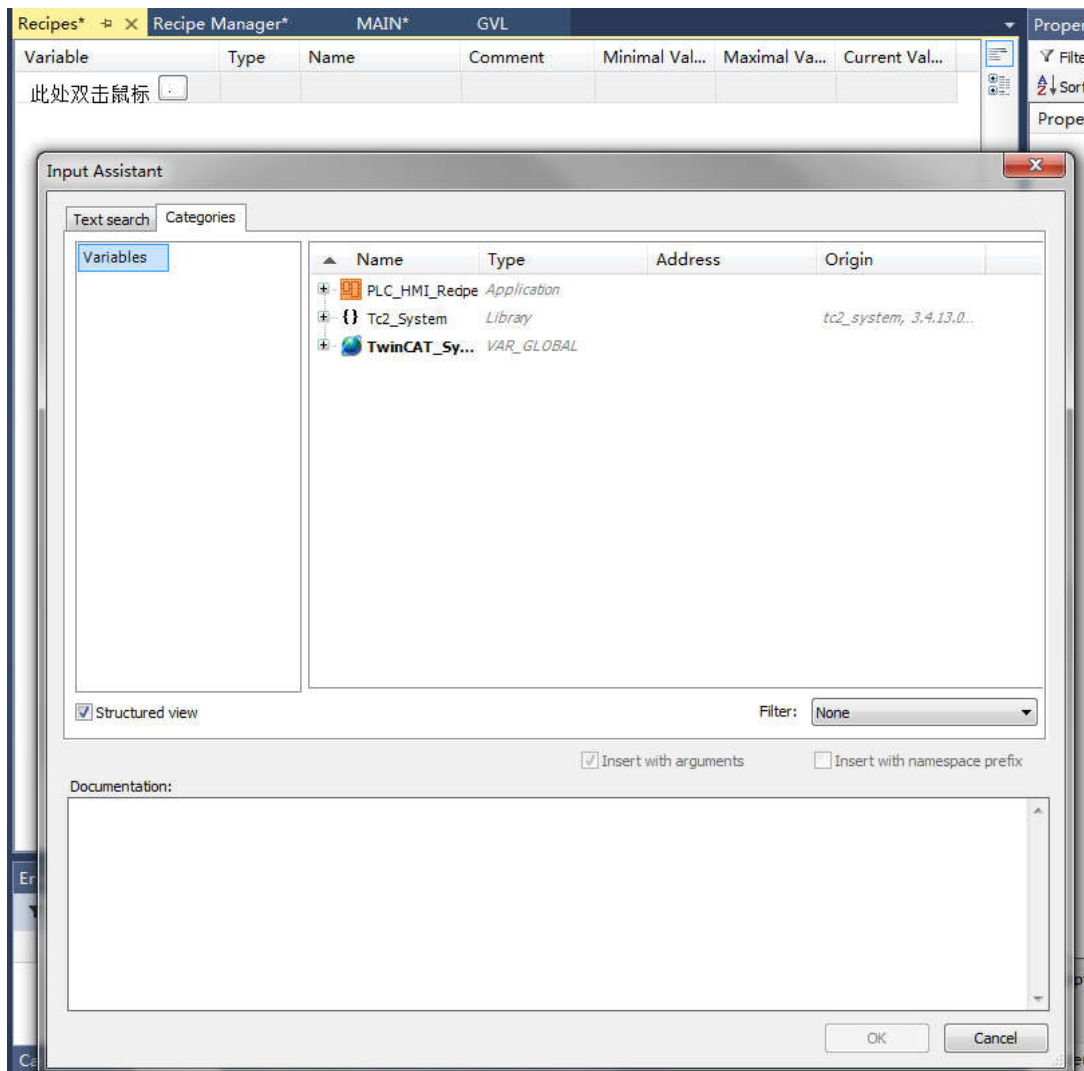
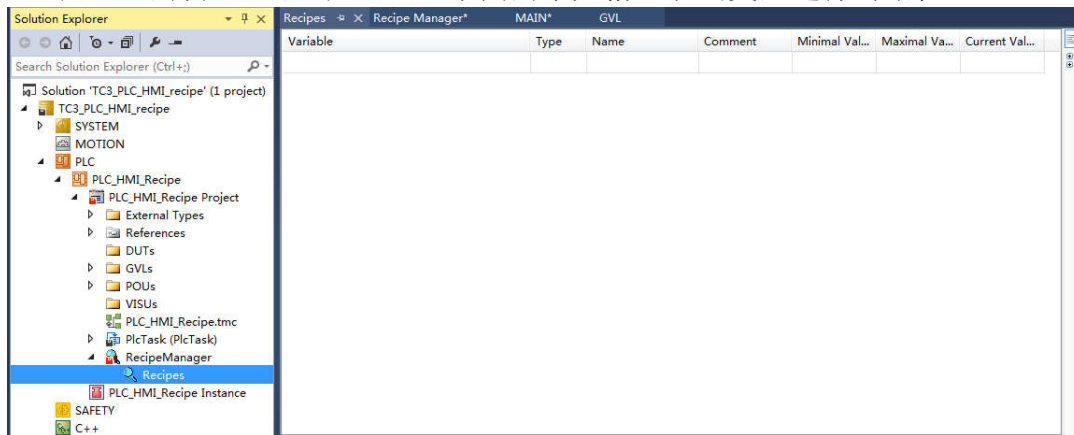
7) 鼠标在 Recipe Manager 上右键—>添加->Recipe Definition



8) 弹出对话框，保持默认名“Recipes”。然后点击‘Open’按钮。

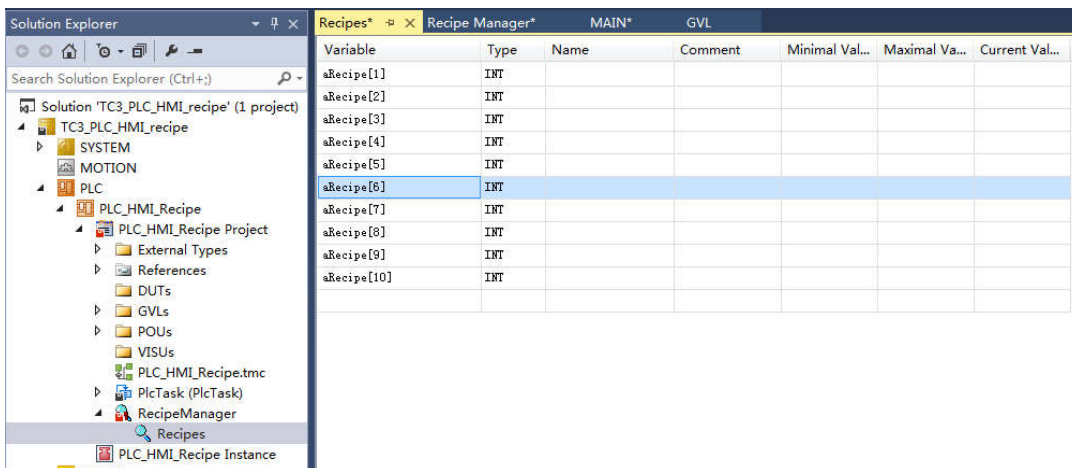
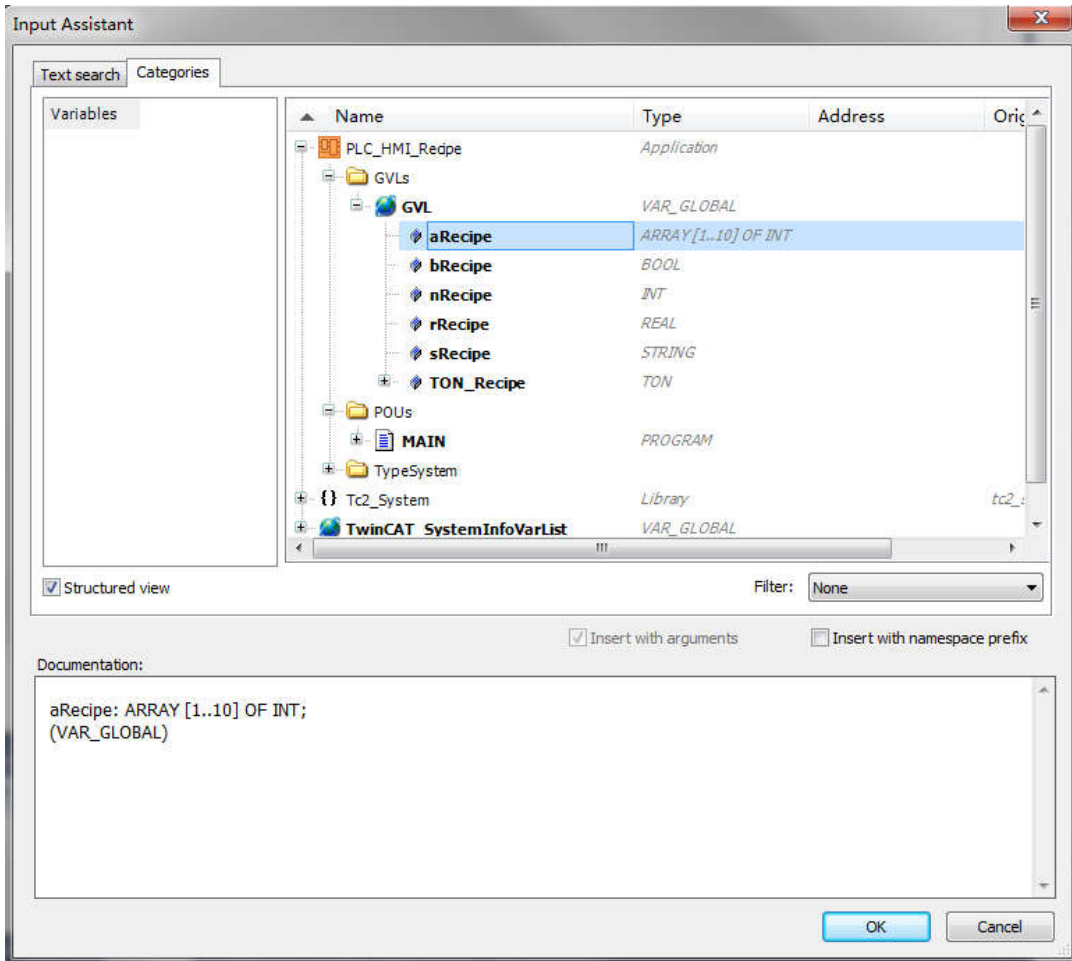


9) 在右边的窗口，双击 **Variable** 下面的单元格，跳出变量选择对话框：





10) 本例选择全局量变名为 **aRecipe** 的数组。可见变量已经添加进来了。可以根据实际的配方功能，继续添加变量。



# BECKHOFF

---

11) 如图可见，局部变量也是可以添加进来的。结构也可以。功能块也可以。非常强大但又很方便。

|                          |        |  |  |  |  |
|--------------------------|--------|--|--|--|--|
| MAIN.bRecipe             | BOOL   |  |  |  |  |
| MAIN.nRecipe             | INT    |  |  |  |  |
| MAIN.rRecipe             | REAL   |  |  |  |  |
| MAIN.aRecipe[1]          | INT    |  |  |  |  |
| MAIN.aRecipe[2]          | INT    |  |  |  |  |
| MAIN.aRecipe[3]          | INT    |  |  |  |  |
| MAIN.aRecipe[4]          | INT    |  |  |  |  |
| MAIN.aRecipe[5]          | INT    |  |  |  |  |
| MAIN.aRecipe[6]          | INT    |  |  |  |  |
| MAIN.aRecipe[7]          | INT    |  |  |  |  |
| MAIN.aRecipe[8]          | INT    |  |  |  |  |
| MAIN.aRecipe[9]          | INT    |  |  |  |  |
| MAIN.aRecipe[10]         | INT    |  |  |  |  |
| MAIN.sRecipe             | STRING |  |  |  |  |
| MAIN.TON_Recipe.IN       | BOOL   |  |  |  |  |
| MAIN.TON_Recipe.PT       | TIME   |  |  |  |  |
| MAIN.TON_Recipe.Q        | BOOL   |  |  |  |  |
| MAIN.TON_Recipe.ET       | TIME   |  |  |  |  |
| MAIN.TON_Recipe.M        | BOOL   |  |  |  |  |
| MAIN.TON_Recipe.Start*** | TIME   |  |  |  |  |

## 三、PLC HMI 画面

### 11) 选择 HMI 画面

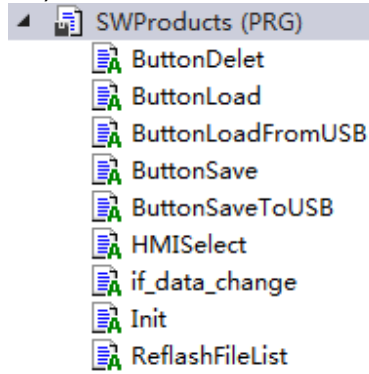
The screenshot displays the Beckhoff software interface. On the left, a project tree is visible under the 'PLC\_HMI\_Recipe' project. The tree structure is as follows:

- PLC\_HMI\_Recipe
  - PLC\_HMI\_Recipe Project
    - External Types
    - References
    - DUTs
    - GVLs
    - POUs
      - 配方功能
        - FB\_DirectoryCopy (FB)
        - FB\_FileCopy (FB)
        - GVL\_Products
        - HMI\_PRODUCTS**
        - RecipeArr
        - RecipeManager
          - Recipes
        - SWProducts (PRG)
        - TextList\_ProductsDiag
      - MAIN (PRG)

The 'HMI\_PRODUCTS' folder is highlighted in the tree. Below the tree, the HMI recipe selection screen is shown. It features a header bar with the number '333'. A list of recipe numbers is displayed, with '123' and '333' visible. To the right of the list are several control buttons: '读取' (Read), '保存' (Save), and '删除' (Delete). Below these buttons is an input field labeled '输入配方名' (Enter recipe name) with the value '798' entered. Further down are buttons for '从USB读取' (Read from USB) and '下载到USB' (Download to USB). At the bottom, there is a '刷新列表' (Refresh list) button.

## 四、程序运行

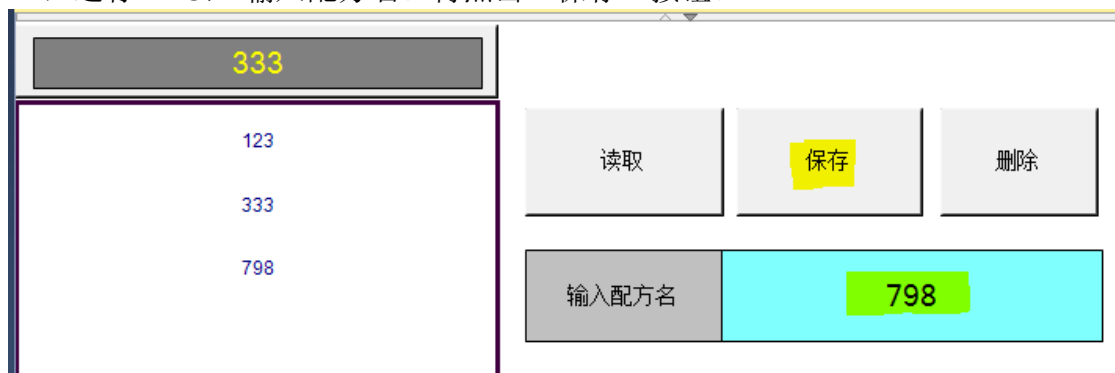
12)HMI 画面直接和程序绑定了，不需要修改程序。封装完成，即拿即用。



13)在“SWProducts”程序里面修改变量：DeviceNum，选择运行程序的控制器操作系统。

```
1 DeviceNum:=1; (*XP/CE操作系统选择*)
2
3 IF vbtLoad OR vbtSave THEN
4     CASE DeviceNum OF
5         1: (*xp*)
6             sDevice:='C:\Products\';//不建议修改保存目录,其他地方也可能要改的
7         2: (*usb*)
8             ;
9         3: (*ce*)
10            sDevice:='\\Hard Disk\Products\';//不建议修改保存目录,其他地方也可能要改的
11    END_CASE;
12 END_IF
```

17) 运行 PLC，输入配方名。再点击“保存”按钮。



可以得到配方保存的文件，如图：

# BECKHOFF

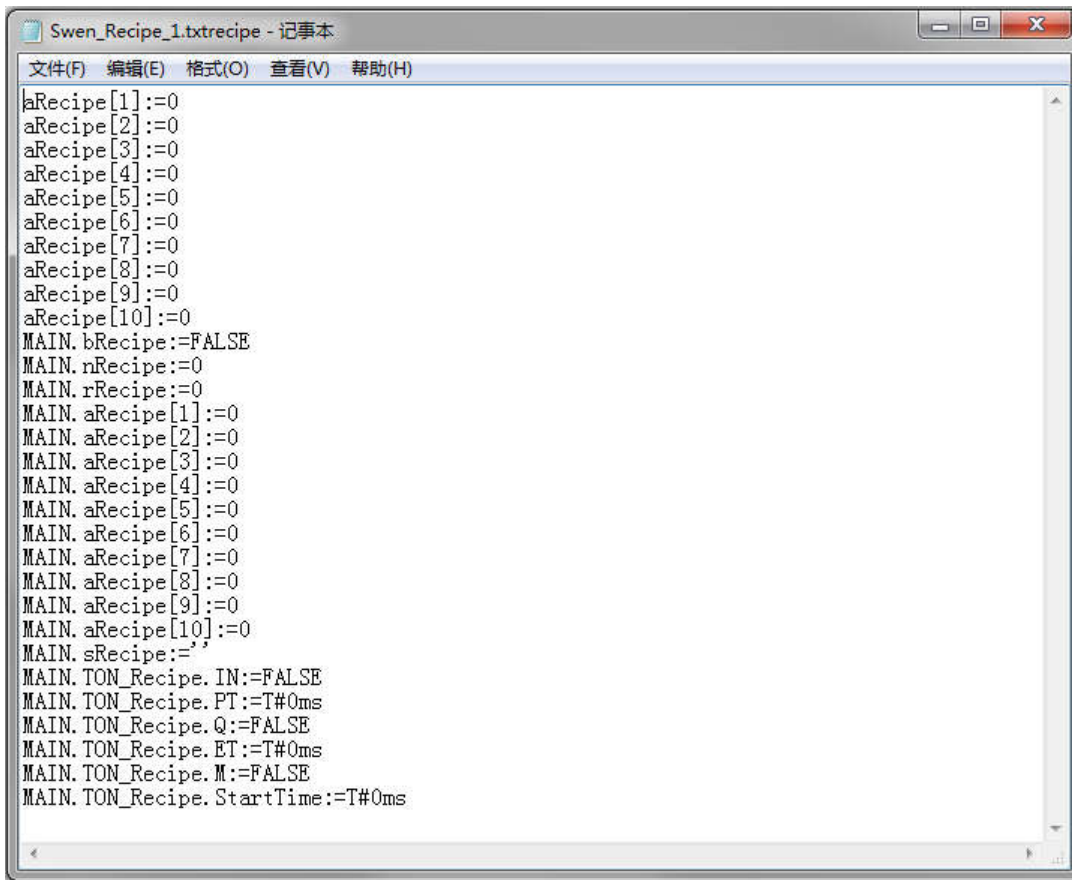


配方保存的路径，为程序选择的目录。一般情况下，不要修改该路径。只需要选择操作系统就可以了。

```
1 DeviceNum:=1; (*XP/CE操作系统选择*)
2
3 IF vbtLoad OR vbtSave THEN
4     CASE DeviceNum OF
5         1: (*xp*)
6             sDevice:='C:\Products\';//不建议修改保存目录,其他地方也可能要改的
7         2: (*usb*)
8             ;
9         3: (*ce*)
10            sDevice:='\Hard Disk\Products\';//不建议修改保存目录,其他地方也可能要改的
11    END_CASE;
12 END_IF
```

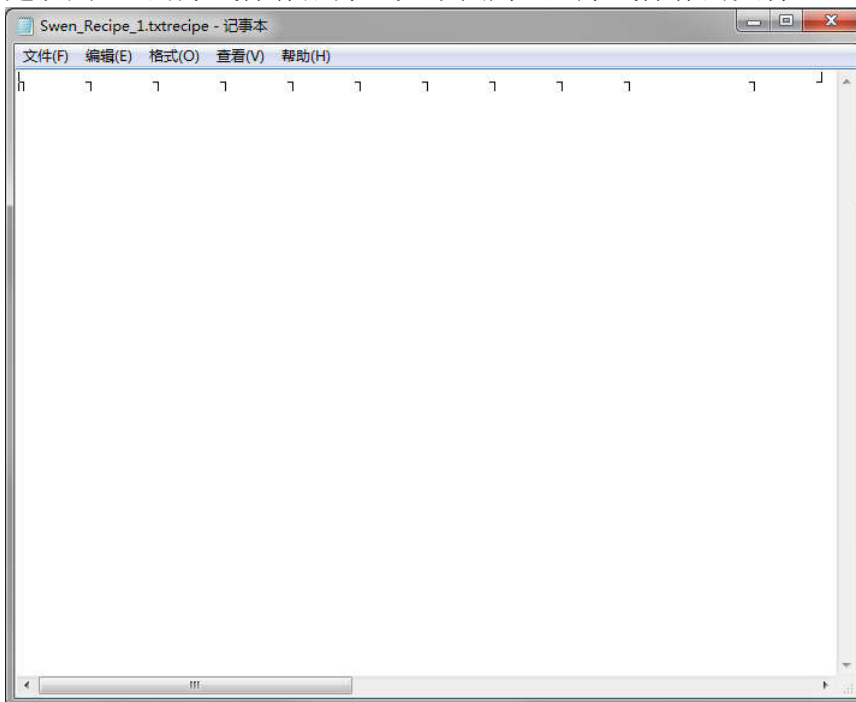
可以用记事本打开文件，可以查看里面的数据。

# BECKHOFF



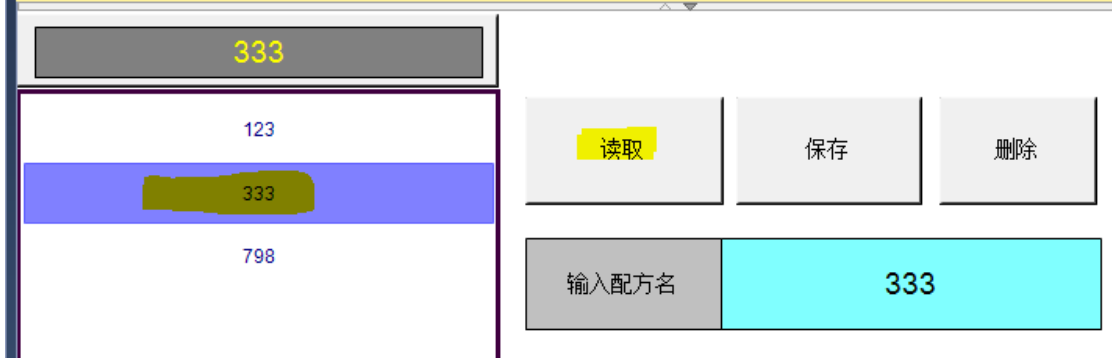
```
Swen_Recipe_1.txtrecipe - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
aRecipe[1]:=0
aRecipe[2]:=0
aRecipe[3]:=0
aRecipe[4]:=0
aRecipe[5]:=0
aRecipe[6]:=0
aRecipe[7]:=0
aRecipe[8]:=0
aRecipe[9]:=0
aRecipe[10]:=0
MAIN.bRecipe:=FALSE
MAIN.nRecipe:=0
MAIN.rRecipe:=0
MAIN.aRecipe[1]:=0
MAIN.aRecipe[2]:=0
MAIN.aRecipe[3]:=0
MAIN.aRecipe[4]:=0
MAIN.aRecipe[5]:=0
MAIN.aRecipe[6]:=0
MAIN.aRecipe[7]:=0
MAIN.aRecipe[8]:=0
MAIN.aRecipe[9]:=0
MAIN.aRecipe[10]:=0
MAIN.sRecipe:='',
MAIN.TON_Recipe.IN:=FALSE
MAIN.TON_Recipe.PT:=T#0ms
MAIN.TON_Recipe.Q:=FALSE
MAIN.TON_Recipe.ET:=T#0ms
MAIN.TON_Recipe.M:=FALSE
MAIN.TON_Recipe.StartTime:=T#0ms
```

如果在第‘6)’步时选择 **bin** 的方式，则不能查看配方里面的数据，显示为乱码。所以建议用 **txt** 的方式保存配方。如下图所示为 **bin** 方式保存的文件：

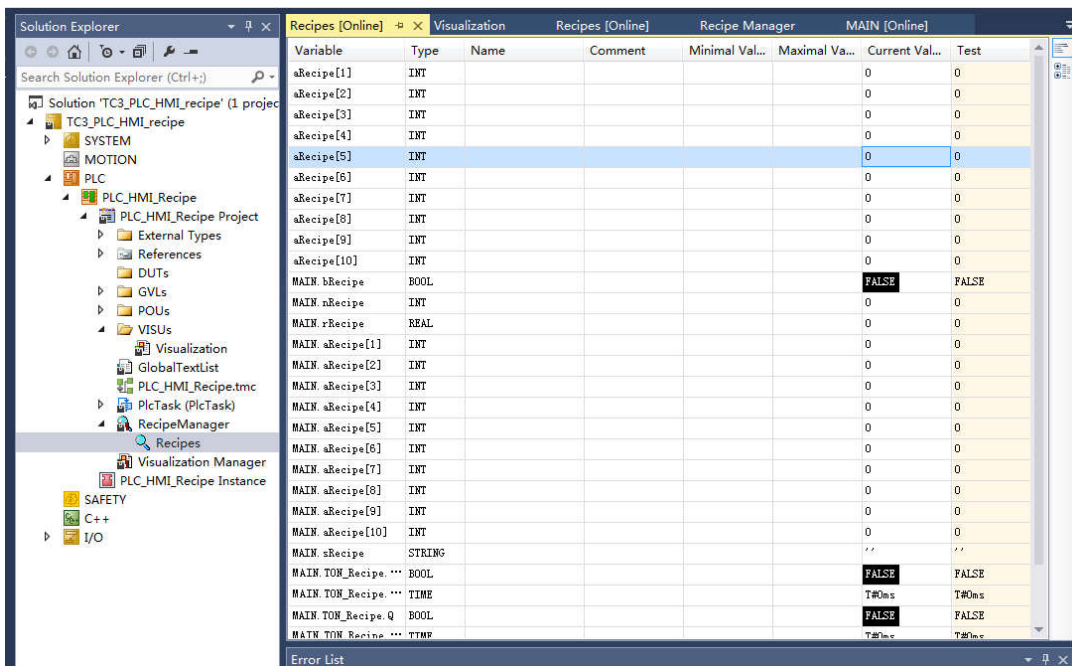


```
Swen_Recipe_1.txtrecipe - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
h      1      1      1      1      1      1      1      1      1
```

18) 在左边的选择框里选择需要的配方，再点击“读取按钮”，可以激活配方。



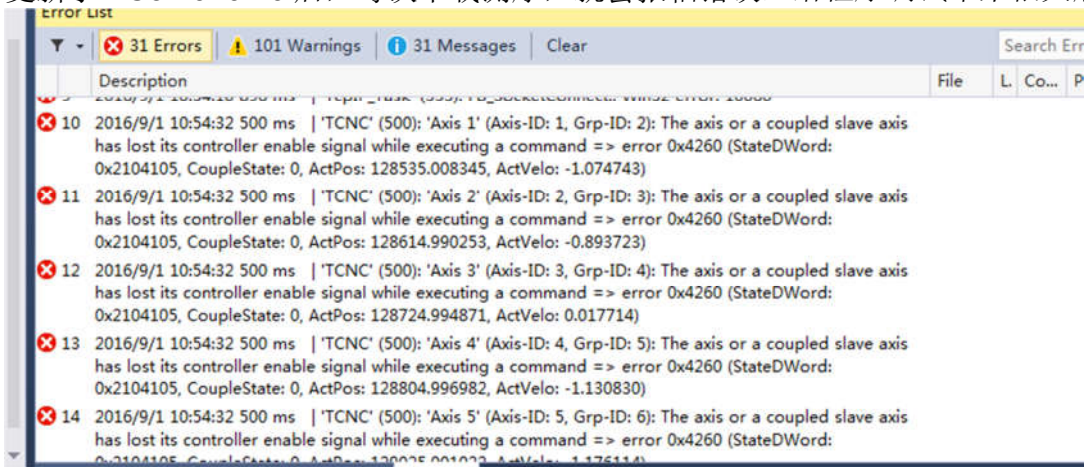
19) 在 Recipe Manager 里面可以监控配方里面的数据。



## 附录：

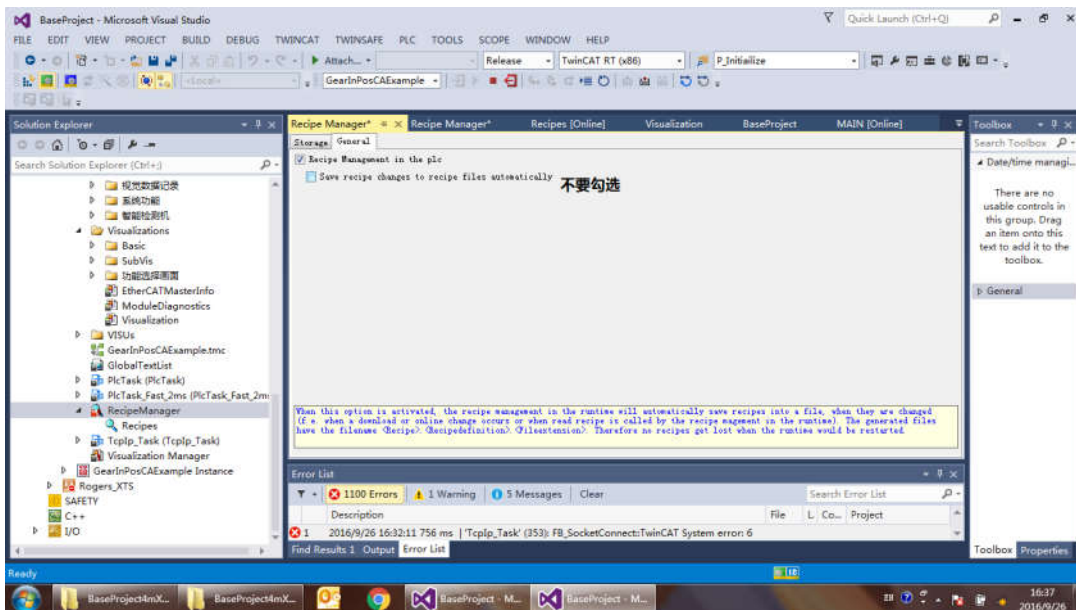
调试笔记一则：

更新了 TC3.4020.10 后，每次下载测序，就会报轴错误，给程序调试带来很大麻烦：




经过努力排查，上网查找资料，发现是 RecipeManager 导致的。需要把自动保存功能关闭。

分析：勾选了自动保存后，每次下载更新程序，都会被认为需要触发自动保存。导致 cpu 资源被占用（该项目的配方数据较多）。最后导致需要正常使用 cpu 的 task 任务超时，导致报错。

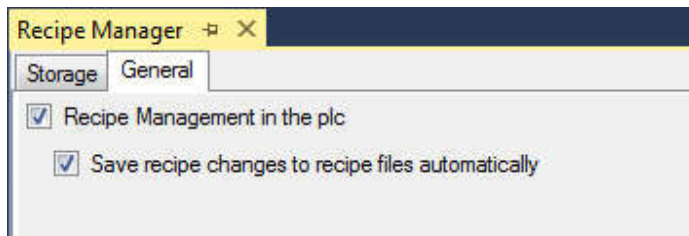




如下面帮助文件里面，我用红色标注的部分：

|   |  |
|---|--|
|  <p><b>CAUTION</b></p> | <p><b>Floating-point numbers in textual recipe files</b></p> <p>Floating-point values (type REAL/LREAL) are stored in textual recipe files in decimal format and in hexadecimal format.</p> <p>If you wish to change a value in the recipe file, edit only the decimal value and remove the hexadecimal entry.</p> |
|---|--|

## Tab "General"



The two options "Recipe Management in the plc" and "Save recipe changes to recipe files automatically" are enabled by default. It is recommended to retain this setting.

### "Recipe management in the plc" option

If the "Recipe Management in the plc" option is enabled, the recipe manager is loaded during the download to the controller. There, it handles the reading and writing of recipes while the application is running on the controller.

If there is no connection to the programming device, the recipe commands (create, read, write, save to file, load from file) are managed via appropriately configured visualization elements or via the "RecipeManCommands" function block from the "RecipeManagement" library, with whose methods the recipes in the application code can be changed.

Disable the "Recipe Management in the plc" option if the recipe manager is not needed on the controller, for example if no recipes are used at runtime, or if the recipes are used only to exchange certain parameters during commissioning, if the programming device is still connected to the controller.

The reading and writing of recipe values then takes place via the default options in the context menu of the recipe definitions.

### "Save recipe changes to recipe files automatically" option

**NOTE! This option is only effective if the recipe manager is loaded to the controller ("Recipe Management in the plc" option is enabled).**

## BECKHOFF

---

If the "Save recipe changes to recipe files automatically" option is enabled, the recipe manager automatically saves the recipes to a file at runtime in the event of changes, i.e. each time a recipe is changed, the stored data are automatically updated at runtime (e.g. during a download, an online change or if the read recipe is called at runtime by the recipe manager).

▪  
Disable the "Save recipe changes to recipe files automatically" option if the storage is to be controlled explicitly by the application.