**1-Second UPS介绍及Persisten变量存储原理**

|  |  |
| --- | --- |
|  | 作者：崔维涛  职务：风电行业 技术工程师  公司：BECKHOFF中国  邮箱：w.cui@beckhoff.com.cn  日期：2024-01-05 |
| **摘 要：**  本文介绍了1-Second UPS功能及使用注意事项，并展开测试了Persistent变量一些属性，最后给出1-Second UPS使用的一般方法。 | |
| **附 件：**   |  |  |  | | --- | --- | --- | | 序 号 | 文件名 | 备注 | |  |  |  | |  |  |  | |  |  |  | |  |  |  | | |
| **历史版本：**   |  |  |  | | --- | --- | --- | | 2024-01-05 | 崔维涛 | 首次创建 | |  |  |  | |  |  |  | |  |  |  | |  |  |  | | |
| **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，如有改动，恕不事先通知，也欢迎您提出改进建议。 | |
| **参考信息：** | |

目 录

[1 1-Second UPS保持文件工作原理 3](#_Toc170461872)

[1.1 关于.wbp及.wb~文件 3](#_Toc170461873)

[1.1.1 文件第一个4字节 4](#_Toc170461874)

[1.1.2 文件第二个4字节 4](#_Toc170461875)

[1.1.3 每个Entry 5](#_Toc170461876)

[1.2 关于FB\_S\_UPS\_CX51x0 7](#_Toc170461877)

[1.3 关于Persistent变量 7](#_Toc170461878)

[2 1-Second UPS使用的限制 7](#_Toc170461879)

[2.1 CX50x0 8](#_Toc170461880)

[2.2 CX51x0 8](#_Toc170461881)

[3 测试项目 8](#_Toc170461882)

[3.1 被测控制器 8](#_Toc170461883)

[3.2 掉电触发FB\_S\_UPS\_CX51x0保存 8](#_Toc170461884)

[3.2.1 目的 8](#_Toc170461885)

[3.2.2 测试过程 9](#_Toc170461886)

[3.2.3 小结 9](#_Toc170461887)

[3.3 程序内使用FB\_WritePersistentData保存 9](#_Toc170461888)

[3.3.1 目的 9](#_Toc170461889)

[3.3.2 测试过程 9](#_Toc170461890)

[3.3.3 小结 10](#_Toc170461891)

[3.4 掉电保持文件内变量保存顺序 10](#_Toc170461892)

[3.4.1 目的 10](#_Toc170461893)

[3.4.2 初始定义 11](#_Toc170461894)

[3.4.3 第一次追加 11](#_Toc170461895)

[3.4.4 第二次追加 12](#_Toc170461896)

[3.4.5 第三次追加 12](#_Toc170461897)

[3.4.6 第四次追加 13](#_Toc170461898)

[3.4.7 小结 14](#_Toc170461899)

[3.5 人为修改.wbp文件 14](#_Toc170461900)

[3.5.2 删除.wbp文件时，保留.wb~ 16](#_Toc170461901)

[3.5.3 删除.wbp和.wb~ 16](#_Toc170461902)

[3.5.4 小结 17](#_Toc170461903)

[4 结论 18](#_Toc170461904)

# 1-Second UPS保持文件工作原理

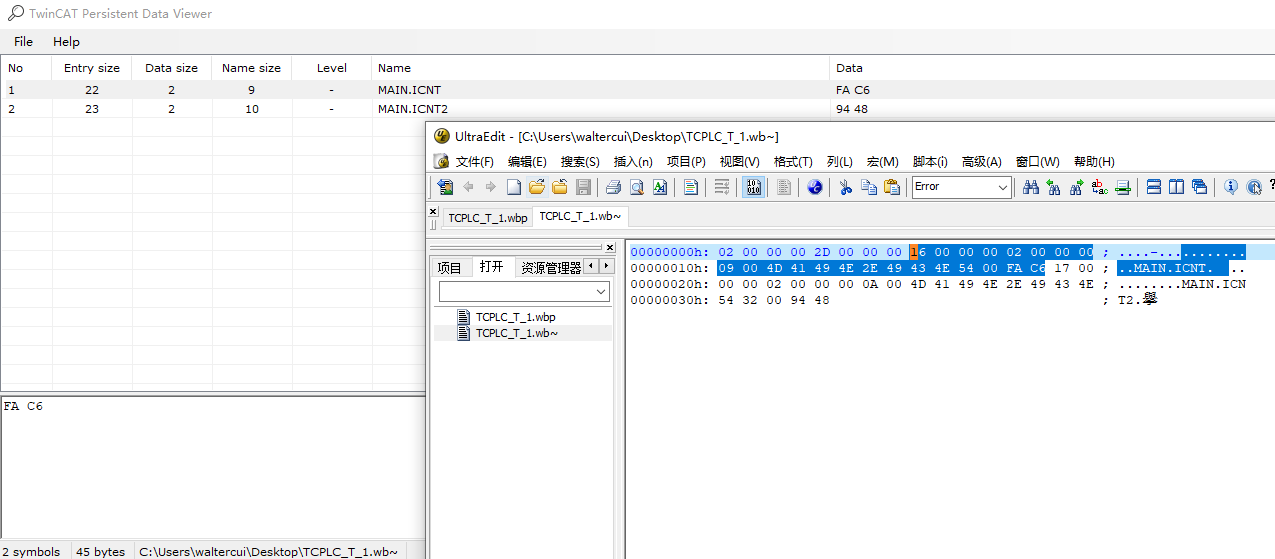
PLC代码内必须每个周期都要调用FB\_S\_UPS\_CX51x0的实例。该FB通过检测供电状态，在主电源掉电的情况下把Persistent类型变量全部写入文件（TCPLC\_T\_x.wbp）中。再次上电时，读取TCPLC\_T\_x.wbp中的数据，初始化到程序内，并备份为TCPLC\_T\_x.wb~，然后把TCPLC\_T\_x.wbp删除。如果TCPLC\_T\_x.wbp文件损坏，则使用TCPLC\_T\_x.wb~内的数据初始化程序内的Persistent变量（需要在注册表中设置不清除无效的Persistent变量）。

## 关于.wbp及.wb~文件

Persistent变量保存的文件为.wbp及.wb~文件。这两种文件的格式相同，.wb~作为.wbp的备份文件使用。两种文件保存的内容如下图所示，可以看到，文件中不仅存储了Persistent变量的值，还包括变量字节数、变量名、变量名字节数等。

图形用户界面, 文本, 应用程序, Word

描述已自动生成



### 文件第一个4字节

图形用户界面, 应用程序

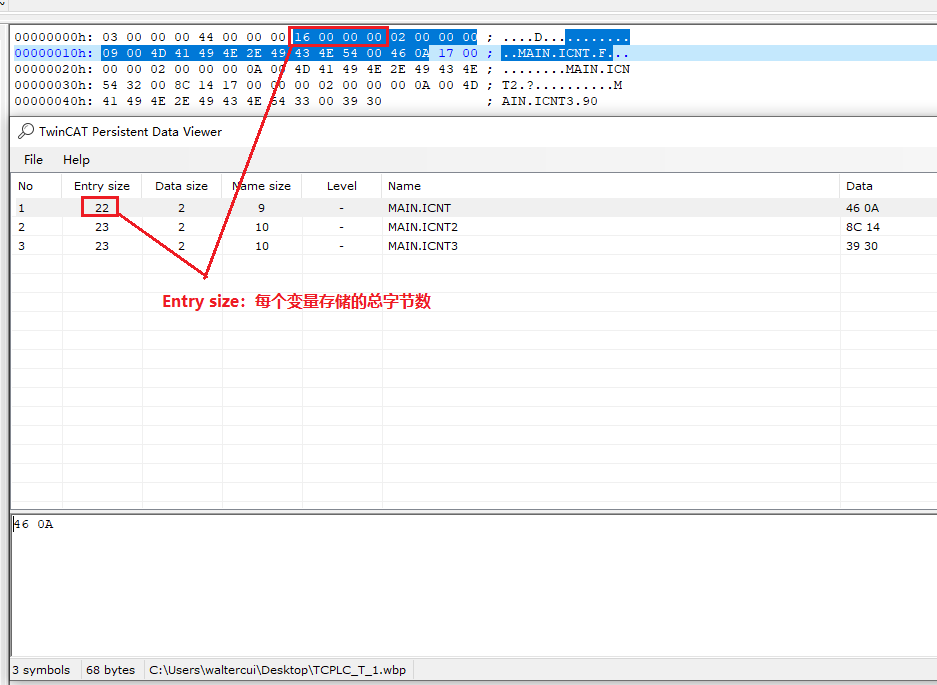
描述已自动生成

### 文件第二个4字节

图形用户界面, 应用程序, 表格

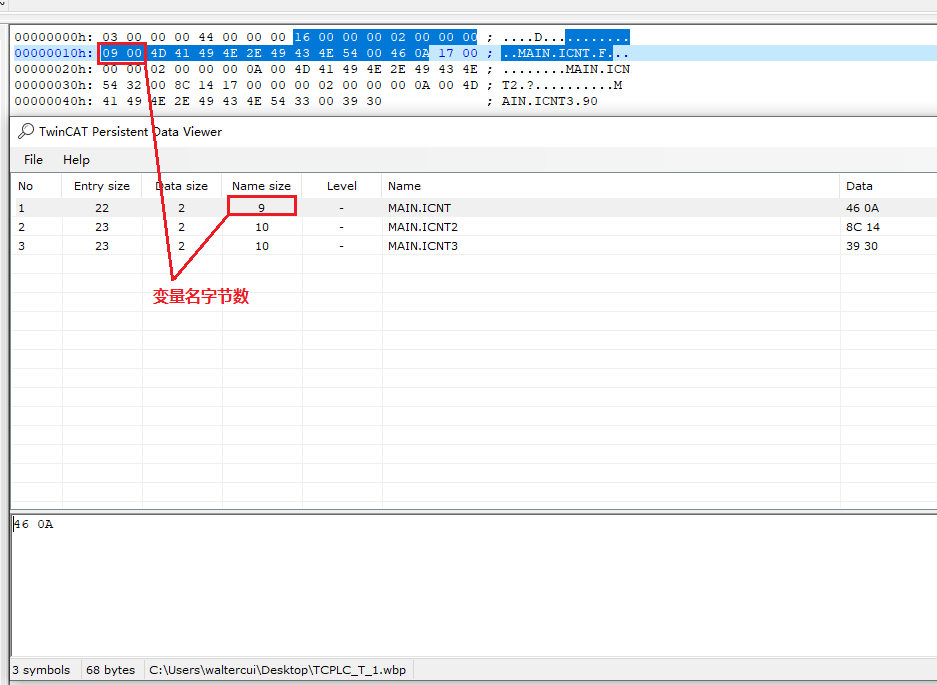
描述已自动生成

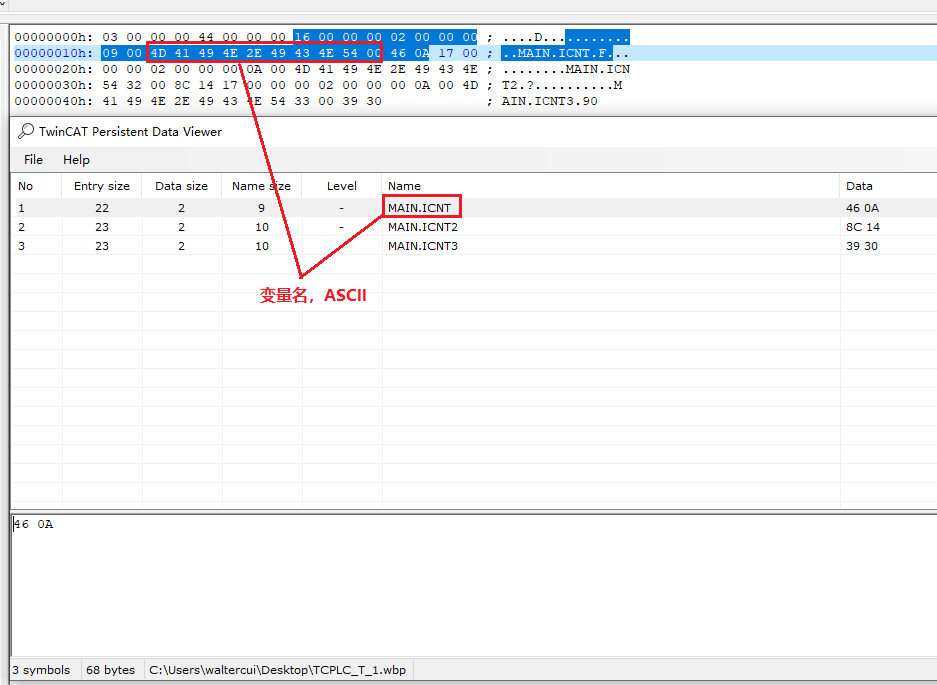
### 每个Entry



图形用户界面, 应用程序

描述已自动生成





图形用户界面, 应用程序

描述已自动生成

## 关于FB\_S\_UPS\_CX51x0

可以使用PLC Control打开TcSUPS\_CX51x0.lib文件，可以看到其执行的逻辑。大概过程如下，检测电源供电状态，当掉电时，如果需要记录Persistent变量，则调用FB\_WritePersistentData的实例来实现将Persistent变量存储到Cfast卡的文件（TCPLC\_T\_x.wbp）中。文件存储完毕则执行关机命令。因此在执行写Persistent变量时，有可能会出错，需要根据错误码去分析具体出错原因。

## 关于Persistent变量

Persistent变量的初始化由TwinCAT来实现，无需PLC代码干预。TwinCAT启动时会自动加载.wbp文件用于初始化Persistent变量，如果文件中的数据无效，则使用Persistent变量定义时的初始值。

# 1-Second UPS使用的限制

手册中已经明确说明，为保证在控制器的寿命周期内，1sUPS都能正常保存数据，1sUPS最大保存1M字节的数据。

## CX50x0

图片包含 图形用户界面

描述已自动生成

## CX51x0

图片包含 应用程序

描述已自动生成

# 测试项目

## 被测控制器

CX5130：2016.2.10生产

图形用户界面, 应用程序

描述已自动生成

## 掉电触发FB\_S\_UPS\_CX51x0保存

### 目的

在调用FB\_S\_UPS\_CX51x0时，使用不同模式，看是否对task执行周期有影响及能否正常保存文件。

### 测试过程

两个Task，每个Task内5M的Persistent变量，仅测试程序。

|  |  |  |
| --- | --- | --- |
| **eUpsMode** | **ePersistentMode** | **效果** |
| eSUPS\_WrPersistData\_Shutdown | SPDM\_2PASS | Task未超时，写完数据后直接关机，关机前Tc使用率从1%升到18% |
| eSUPS\_WrPersistData\_NoShutdown | SPDM\_2PASS | Task未超时，写完数据后进入等待电源恢复状态，关机前Tc使用率从1%升到18% |
| eSUPS\_CheckPowerStatus | —— | 不涉及写Persistent变量 |
| eSUPS\_ImmediateShutdown | —— | 不涉及写Persistent变量 |
| eSUPS\_WrPersistData\_Shutdown | SPDM\_VAR\_BOOST | Task未超时，写完数据后直接关机，关机前Tc使用率从1%升到31% |
| eSUPS\_WrPersistData\_NoShutdown | SPDM\_VAR\_BOOST | Task未超时，写完数据后进入等待电源恢复状态，关机前Tc使用率从1%升到19% |

### 小结

1. 使用SPDM\_2PASS和SPDM\_VAR\_BOOST时效果一样，都能保存数据到.wbp文件内，且任务都没有超时现象。
2. 默认使用的参数eSUPS\_WrPersistData\_Shutdown + SPDM\_2PASS是最佳参数组合，对CPU符合影响不大且能有效使用1sUPS。

## 程序内使用FB\_WritePersistentData保存

### 目的

在正常运行程序中能否正常调用FB\_WritePersistentData，且修改其不同模式，看是否对task执行周期有影响及能否正常保存文件。

### 测试过程

两个Task，每个Task内5M的Persistent变量，仅测试程序。

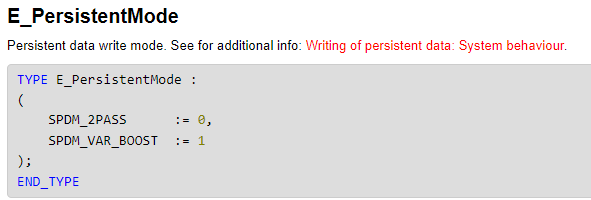
|  |  |  |
| --- | --- | --- |
| **模式** | **Task是否超时** | **CPU使用率** |
| SPDM\_2PASS | 否 | Tc使用率从1%升到31% |
| SPDM\_VAR\_BOOST | 否 | Tc使用率从1%升到18% |

不同数据量下，FB\_WritePersistentData的执行周期，仅测试程序。

|  |  |  |
| --- | --- | --- |
| **任务周期** | **数据大小** | **执行周期** |
| 10ms | 10M | 13 |
| 5M | 7 |
| 1M | 3 |

### 小结

1. 程序可以调用FB\_WritePersistentData的实例将所有Persistent变量保存到.wbp文件内。
2. TwinCAT在正常进入Stop状态时也会自动保存所有Persistent变量保存到.wbp文件内，无需调用任何功能块。
3. 使用SPDM\_2PASS和SPDM\_VAR\_BOOST时效果一样，都能保存数据到.wbp文件内，且任务都没有超时现象。但是，在同样数量的Persistent条件下，使用SPDM\_2PASS要比SPDM\_VAR\_BOOST对CPU的负荷占用少。



图片包含 表格

描述已自动生成

## 掉电保持文件内变量保存顺序

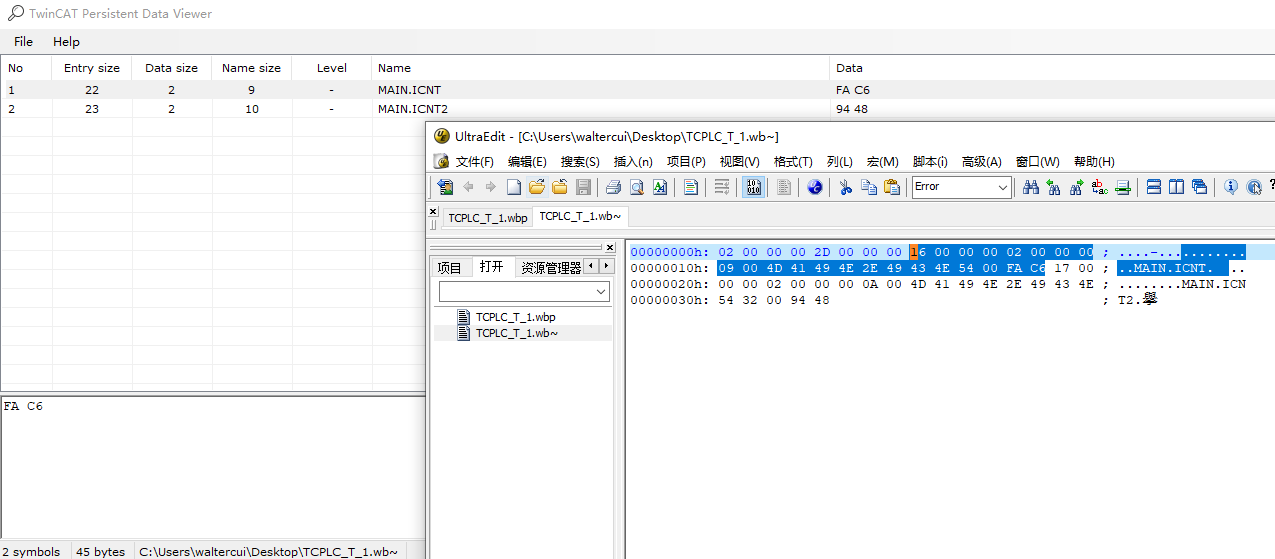
### 目的

测试程序中变量定义顺序是否会影响其在.wbp中的存储顺序。

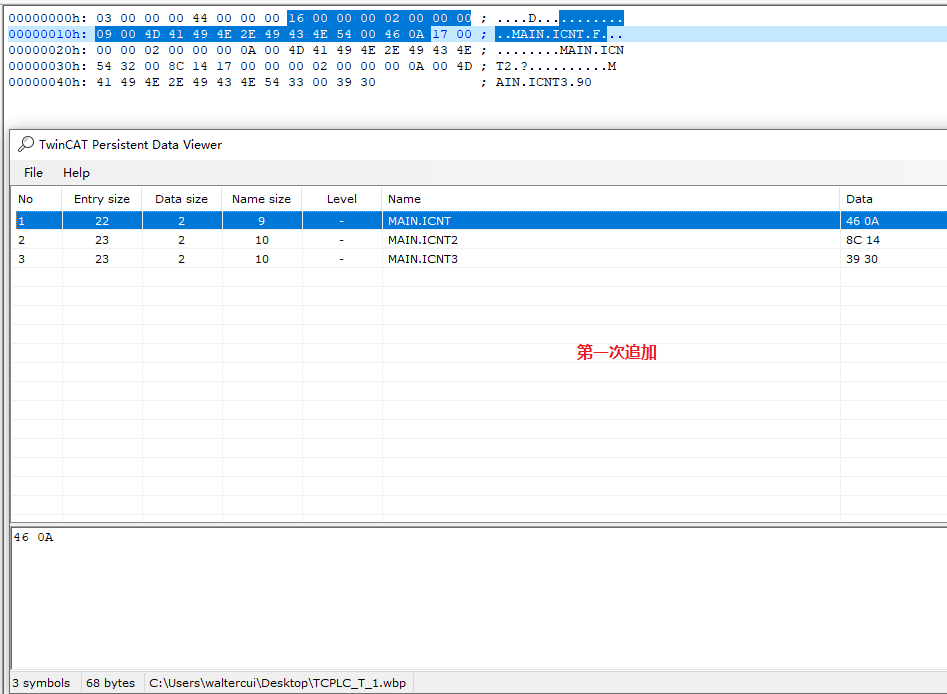
图片包含 文本

描述已自动生成

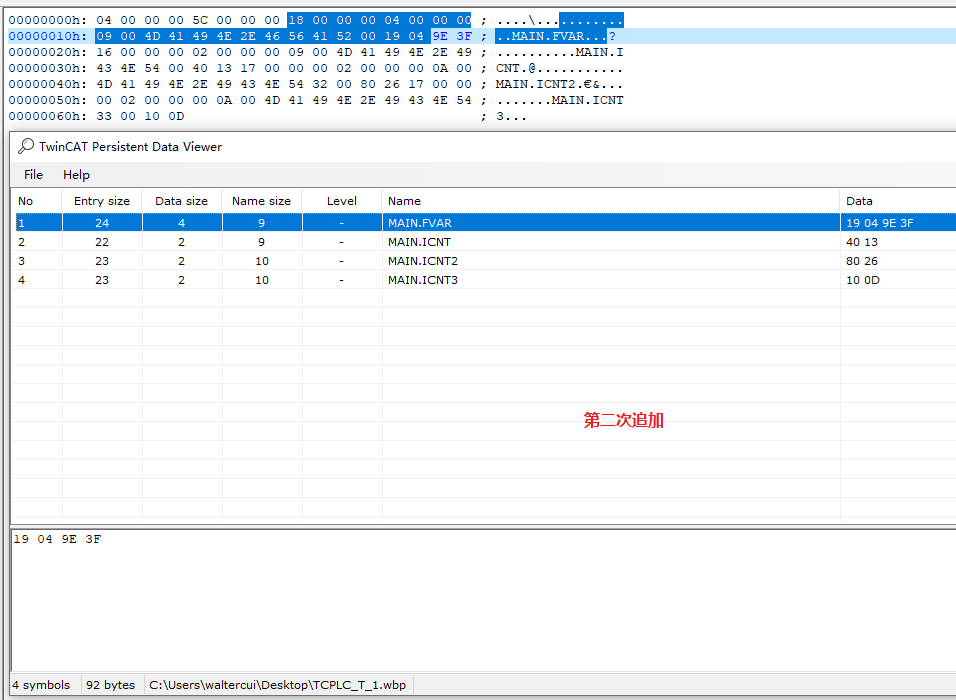
### 初始定义



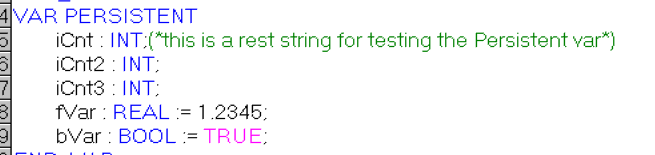
### 第一次追加

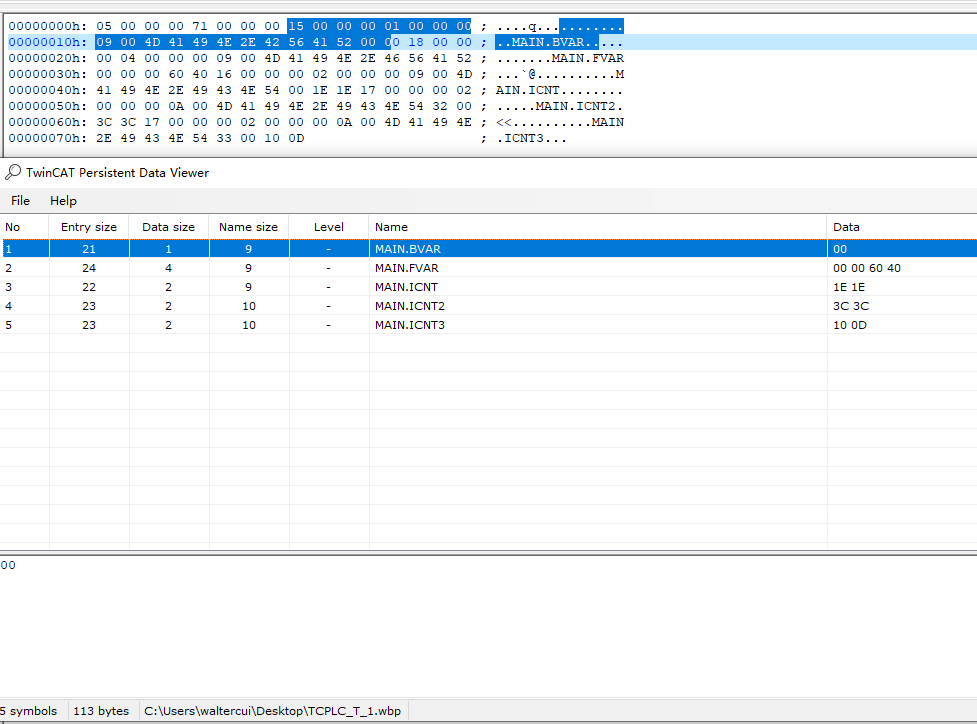


### 第二次追加



### 第三次追加





### 第四次追加

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图片包含 表格

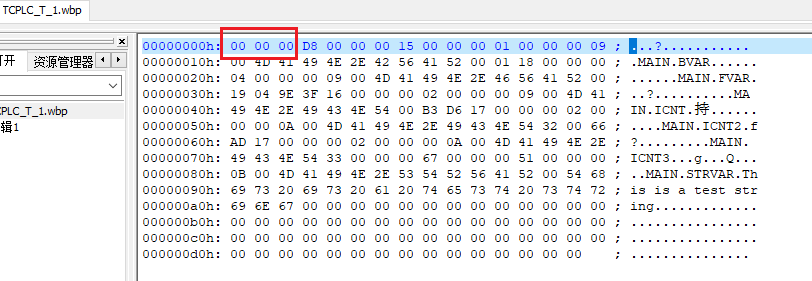
描述已自动生成

### 小结

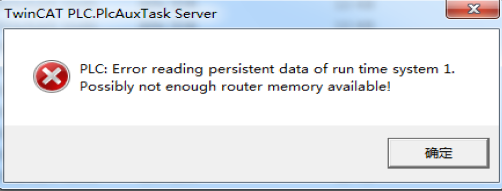
1. Persistent变量存储时是按照内部既定顺序存储，不是按照变量定义顺序来存储。
2. 每个变量存储的空间不仅仅是变量值本身，还包括其它信息。
3. 文件头的位置已经记录了文件内变量数量及所占字节数（此信息可用于对.wbp文件校验：一旦实际读到的字节数与文件头部记录值不一致，则表示文件损坏）。

## 人为修改.wbp文件

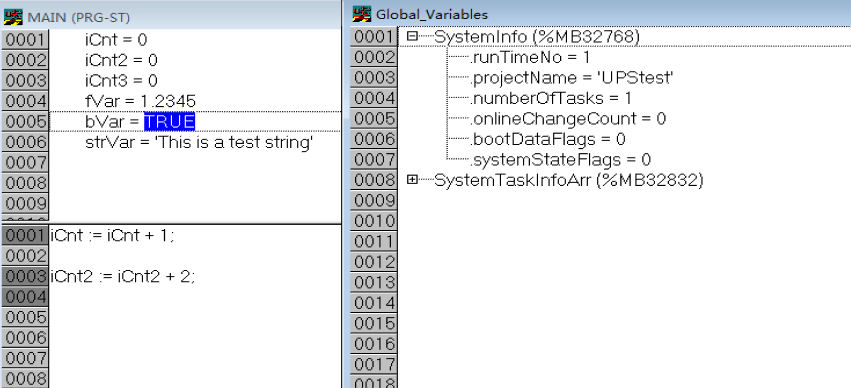
#### 删除文件中的变量个数



报如下错误。



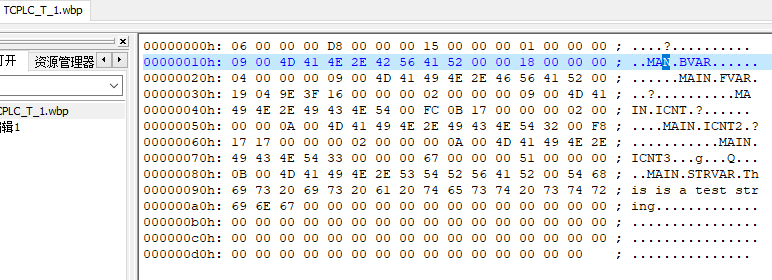
这个可以理解为虽然删除了文件的第一个字节，但是程序是按照文件内的数据顺序来读取，也就是当前程序按00 00 00 D8个变量来读取文件中的内容，最终导致报出上面的错误。该错误是在TwinCAT启动过程中报出的。最终效果如下



所有Persistent变量值变成代码给的初值。

#### 删掉变量名的描述中的某个字节

删掉了MAIN中的I



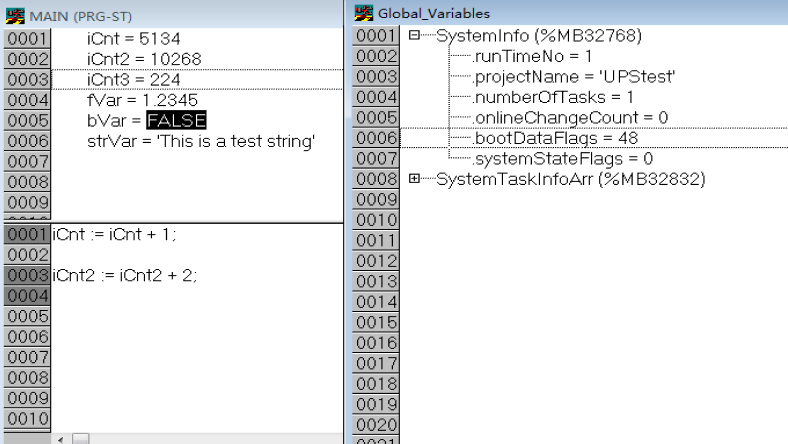
效果如下

表格

描述已自动生成

所有Persistent变量值变成代码给的初值。

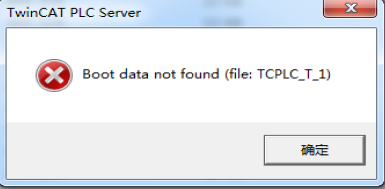
### 删除.wbp文件时，保留.wb~



TwinCAT使用.wb~文件对所有Persistent变量初始化。

### 删除.wbp和.wb~

TwinCAT启动时报错如下



表格

描述已自动生成

所有Persistent变量值变成代码给的初值。

### 小结

1. 如果是改变Data值，那么.wbp文件会被正常读取。
2. 如果改变文件内的字节数，如删掉文件中一个字节，那么程序在初始化时即认为.wbp文件损坏，整个.wbp文件内容被丢弃，TwinCAT使用Persistent变量的初始值。
3. 如果.wbp文件不存在，则启用.wb~文件内的数据（需注册表内设置支持）。
4. 如果.wbp和.wb~文件都不存，则TwinCAT会弹窗报错，并使用Persistent变量的初始值。
5. Persistent变量无效 = .wbp文件不存在。

# 结论

1. Persistent变量在TwinCAT停止时会自动保存到.wbp文件内，重启后会自动初始化，无需调用任何FB。因此，重启CE系统同样会触发上述功能。但是该过程是TwinCAT内部控制，耗时无法确定。
2. 在电源掉电时使用控制器对应的1sUPS的FB，其会自动调用FB\_WritePersistentData的功能块来实现Persistent变量的保存。1sUPS能支撑的时间不确定时，Persistent变量可能无法正常保存。
3. 在程序正常执行过程中调用FB\_WritePersistentData即能实现.wbp文件的正常保存。
4. 使用SPDM\_2PASS和SPDM\_VAR\_BOOST时效果一样，都能保存数据到.wbp文件内，且任务都没有超时现象。但是，在同样数量的Persistent条件下，使用SPDM\_2PASS要比SPDM\_VAR\_BOOST对CPU的负荷占用少。
5. 默认使用的参数eSUPS\_WrPersistData\_Shutdown + SPDM\_2PASS是最佳参数组合，对CPU符合影响不大且能有效使用1sUPS。
6. Persistent变量存储时是按照内部既定顺序存储，不是按照变量定义顺序来存储。
7. 每个变量存储的空间不仅仅是变量值本身，还包括其它信息。因此计算1M的掉电保持文件大小是根据.wbp文件来确定的。
8. 文件头的位置已经记录了文件内变量数量及所占字节数。TwinCAT在初始化时，读到的字节数与文件头部记录值不一致，会直接丢弃.wbp文件中的数据，直接使用程序中的初值。
9. 如果.wbp文件不存在，则启用.wb~文件内的数据（需注册表内设置支持）。
10. 如果.wbp和.wb~文件都不存，则TwinCAT会弹窗报错，并使用Persistent变量的初始值。
11. Persistent变量无效 = .wbp文件不存在。
12. 在默认设置下，控制器上电即给1sUPS的电容充电，且一次正常启动过程（CE系统，TwinCAT正常启动）后，1sUPS即能正常工作，可以写入10M数据（此结论仅代表被测控制器在办公室环境下测得，实际性能以手册为准）。
13. 在1sUPS的电容电量不足的情况下，.wbp文件无法正常生成。此时再上电TwinCAT加载.wb~文件。

**上海（ 中国区总部）**

中国上海市静安区汶水路 299 弄 9号（市北智汇园）

电话: 021-66312666

**北京分公司**

北京市西城区新街口北大街 3 号新街高和大厦 407 室

电话: 010-82200036 邮箱: beijing@beckhoff.com.cn

**广州分公司**

广州市天河区珠江新城珠江东路32号利通广场1303室

电话: 020-38010300/1/2 邮箱: guangzhou@beckhoff.com.cn

**成都分公司**

成都市锦江区东御街18号 百扬大厦2305 室

电话: 028-86202581 邮箱: chengdu@beckhoff.com.cn

|  |  |
| --- | --- |
| 请用微信扫描二维码  通过公众号与技术支持交流 | 倍福官方网站：  https://www.beckhoff.com.cn  在线帮助系统：  https://infosys.beckhoff.com/index\_en.htm |
| 倍福虚拟学院：  https://tr.beckhoff.com.cn/ |
| 招贤纳士：job@beckhoff.com.cn  技术支持：support@beckhoff.com.cn  产品维修：service@beckhoff.com.cn  方案咨询：sales@beckhoff.com.cn |
|  |