**TwinCAT IoT及Analytics用于设备健康管理**

|  |  |
| --- | --- |
|  | 作者：陈利君  职务：华南区 技术工程师  邮箱：l.chen@beckhoff.com.cn  日期：2021-09-14 |
| **摘 要：**  自动化设备的健康管理，其机制就是采集生产设备的健康状态，进行存储和分析。不同行业设备的健康数据、分析工具、健康指标可能会不同，但数据的采集、传输和存储方法是相通的——自动化设备的健康数据由一台或多台PLC提供，但健康状态分析通常在局域网内或者Internet上的另一台PC上进行，数据的提供方与使用方经由IoT技术来连接。  TwinCAT提供两种基于IoT技术的设备健康管理方案：一种是TwinCAT　3　Analytics套件，另一种是PLC自由编程实现IoT数据通讯，二者都采用MQTT协议。实际上这两种方案可以采集任何工艺数据，供用户后期使用。  以上代码由万品雷从自己的机器学习项目中抽取而来，本文仅是测试记录，也可作为代码配套说明。 | |
| **附 件：**   |  |  |  | | --- | --- | --- | | 序 号 | 文件名 | 备注 | | 1 | TwinCATAnalyticsFileSaveLocal | 作者：万品雷 | | 2 | AnalyticsIotMQTT | 作者：万品雷 | | 3 | Analytics Data Collection.pptx | 作者：万品雷 | | 4 |  |  | | |
| **历史版本：**   |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  | | |
| **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 概述 3](#_Toc82512308)

[1.1 设备健康管理方案 3](#_Toc82512309)

[1.2 方式A：TcAnalytic工具 4](#_Toc82512310)

[1.2.1 数据采集 4](#_Toc82512311)

[1.2.2 数据后处理 4](#_Toc82512312)

[1.3 方式B：PLC写代码通过IoT存取数据 4](#_Toc82512313)

[1.3.1 数据采集 4](#_Toc82512314)

[1.3.2 数据后处理 4](#_Toc82512315)

[2 配置方式：Analytics Iot MQTT 5](#_Toc82512316)

[2.1 准备工作 5](#_Toc82512317)

[2.1.1 控制器侧 5](#_Toc82512318)

[2.1.2 MQTT服务侧 5](#_Toc82512319)

[2.1.3 数据记录存储侧 5](#_Toc82512320)

[2.1.4 Analytic数据后处理侧 5](#_Toc82512321)

[2.2 数据采集的配置 6](#_Toc82512322)

[2.2.1 如果单机本地直接采集 6](#_Toc82512323)

[2.2.2 如果通过MQTT服务器 10](#_Toc82512324)

[2.3 数据记录和存储 12](#_Toc82512325)

[2.4 Analytics 数据后处理 16](#_Toc82512326)

[2.4.1 方法一：用TC3 Scope导出 17](#_Toc82512327)

[2.4.2 方法二：用TC3 Scope View 18](#_Toc82512328)

[3 PLC代码方式：TwinCATAnalyticsFileSaveLocal 20](#_Toc82512329)

[3.1 准备工作 20](#_Toc82512330)

[3.2 项目：TcMLFrameWork 20](#_Toc82512331)

[3.2.1 引用库：References 21](#_Toc82512332)

[3.2.2 自定义结构体：ST\_UserData 21](#_Toc82512333)

[3.2.3 自定义功能块：FBs 21](#_Toc82512334)

[3.2.4 全局变量：GVL\_Logger 23](#_Toc82512335)

[3.2.5 全局变量：GVL\_test 23](#_Toc82512336)

[3.2.6 全局变量：GVL 23](#_Toc82512337)

[3.2.7 程序：Main 23](#_Toc82512338)

[3.2.8 程序：PRG\_Logger 23](#_Toc82512339)

[3.2.9 程序：PRG\_Client 25](#_Toc82512340)

[3.3 执行结果 26](#_Toc82512341)

[3.3.1 生成和发送数据 26](#_Toc82512342)

[3.3.2 接收数据 26](#_Toc82512343)

TwinCAT IoT及Analytics用于设备健康管理

By LizzyChen 2021.06.25

* + 资源：来自倍福中国STS组

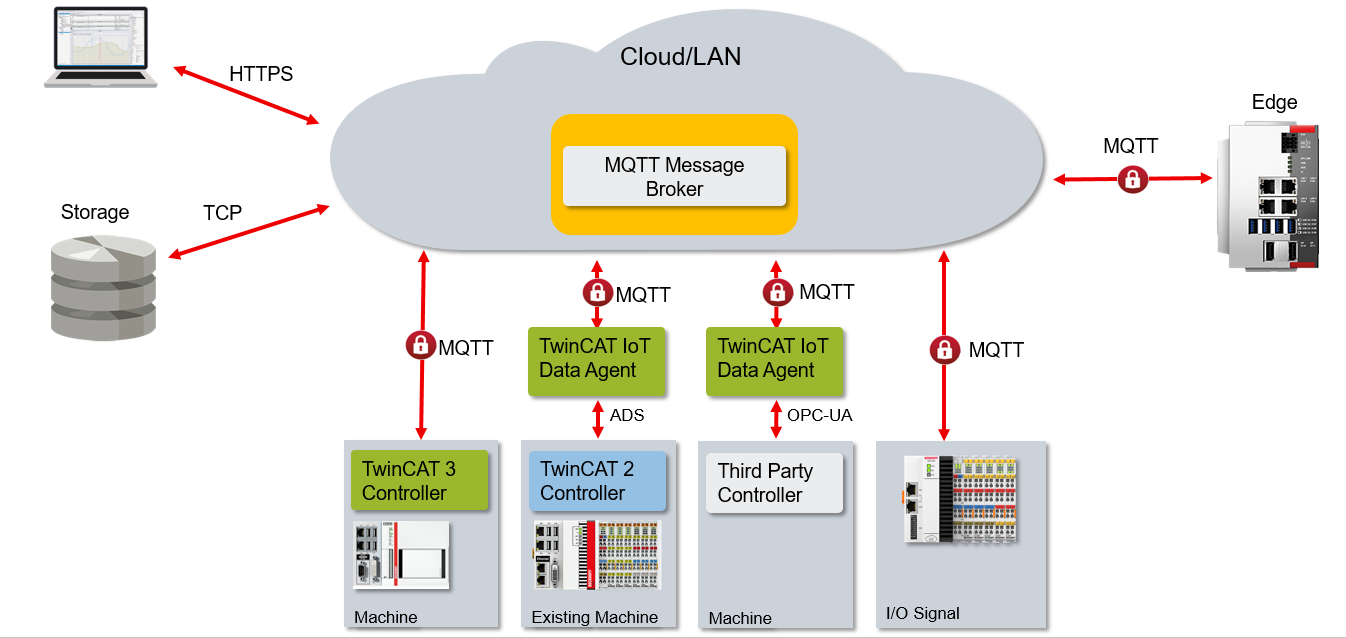


# 概述

## 设备健康管理方案

自动化设备的健康管理，其机制就是采集生产设备的健康状态，进行存储和分析。不同行业设备的健康数据、分析工具、健康指标可能会不同，但数据的采集、传输和存储方法是相通的——自动化设备的健康数据由一台或多台PLC提供，但健康状态分析通常在局域网内或者Internet上的另一台PC上进行，数据的提供方与使用方经由IoT技术来连接。

TwinCAT提供两种基于IoT技术的设备健康管理方案：一种是TwinCAT　3　Analytics套件，另一种是PLC自由编程实现IoT数据通讯，二者都采用MQTT协议。实际上这两种方案可以采集任何工艺数据，供用户后期使用。如图所示：



​上图中，TwinCAT Controller或者Data Agent与MQTT Message Broker建立了MQTT数据流之后：

1. TwinCAT Storage可以引用这个数据流，把数据源源不断地“导入”本地进行存储；
2. 本地存储的文件，又可以在本机或者复制到其它电脑；
3. 其它电脑可以通过TwinCAT Messurement做曲线显示或者通过TwinCAT Analytics Workbench做离线分析；
4. TwinCAT HMI则提供控件和方法可以“实时”显示数据源的变化情况；
5. 边缘设备上运行TwinCAT Analytics Runtime，可以进行在线分析。

不是每个应用都必须用到上述所有功能，本文介绍最简单也最容易实施的应用，就只包含前3点：创建数据流、本地存储和曲线显示。

## 方式A：TcAnalytic工具

详见第2章

### 数据采集

#### DataLogger缓存文件的存储

##### 单机本地直接采集

这是最简单的情况，缓存文件直接就存在PLC的启动目录“C:\TwinCAT\3.1\Boot\Analytics”

通常是倍福IPC上安装TwinCAT XAE，提供数据的PLC程序在IPC上运行，同时存储和分析数据也在这台IPC上执行。

##### MQTT存在服务器上面

这是常用的能发挥MQTT优势的用法，PLC需要连接外网，把临时数据送到指定的Message Broker服务器上。这个服务器可能在局域网，也可能在Internet上。

PLC上不必安装XAE和分析工具，所以对控制器的要求不高，嵌入式PC就可以满足。

#### StorageProvider指定时间段的数据记录

最终需要的数据是指定时间段内特定变量的数据记录。

如果只有DataLogger而没有StorageProvider，按默认配置就只能保存最近5秒的数据。

### 数据后处理

数据采集是为了使用，如果不对数据做后处理，那就没必要采集了。

用配置方式生成的数据，使用特殊的文件格式（.tay），倍福提供数据后处理的系列工具（TC3ScopeExportTool.exe; TC3 XAE的Messurement），用于图形化显示、导出数据、应用分析等。

后处理就是在Scope View中装载数据记录文件（.tay），显示波形曲线，进行各种分析。

也可以导出成.csv或者.txt文件，由用户自定义的第三方工具进行处理。

## 方式B：PLC写代码通过IoT存取数据

### 数据采集

详见第3章。

#### 产生数据侧

PLC把产生的数据上传到MQTT。

除了引用IoT的库，其它都由用户自定义。

#### 存储数据侧

PLC从MQTT Broker获取数据，然后写入csv文件，或者写入数据库。

除了引用IoT的库，其它都由用户自定义。

### 数据后处理

根据PLC存储的数据格式自行定义，比如后续如何分析csv文件，或者如何从数据库提取数据进行分析等。

# 配置方式：Analytics Iot MQTT

## 准备工作

### 控制器侧

授权：控制器侧TF3500 | TC3 Analytics Logger

安装包：已经包含在XAR或XAE中，不需要单独安装

控制器：CX5130，WES7，TC3.1.4024.15(XAR)

### MQTT服务侧

即在MQTT服务器上缓存Analytics数据的功能。如果在云上安装MQTT服务器，Analytics临时中转数据就存在云上。如果在局域网某台PC上安装MQTT服务器，Analytics数据就存在局域网这台PC上。

常用的测试工具是在云服务器或者某台PC上安装mosquitto，使之成为MQTT服务器：

安装包：mosquitto-1.4.14-install-win32.exe，

### 数据记录存储侧

MQTT服务器上只是存储临时数据文件，如果需要记录和存储数据，需要IPC上安装TC3 XAE和TF3520 Analytics StorageProvider：

授权：TF3520 Analytics Storage Provider，

安装包：TF3520-Analytics-Storage-Provider.exe

### Analytic数据后处理侧

做数据后处理的IPC上也要安装TwinCAT 3 XAE和Analytics Engineering

安装包：TC3-Measurement-Full.exe，

*提示：安装时要勾选TE35xx Analytics Engineering，默认是不选中此项的*

#### 方法一：TC3ScopeExportTool.exe

执行文件：

*C:\TwinCAT\Functions\TF3300-Scope-Server\TC3ScopeExportTool.exe*

*C:\TwinCAT\Functions\TE130X-Scope-View\TC3ScopeExportTool.exe*

#### 方法二：TC3 Scope View（TC3 Messurement Project）

TC3 Scope View自带功能

#### 其它

授权：TE3500 | TC3 Analytics Workbench

TE3520 | TC3 Analytics Service Tool

*提示：TE35xx****也有7天试用版授权***

## 数据采集的配置

### 如果单机本地直接采集

#### PLC程序

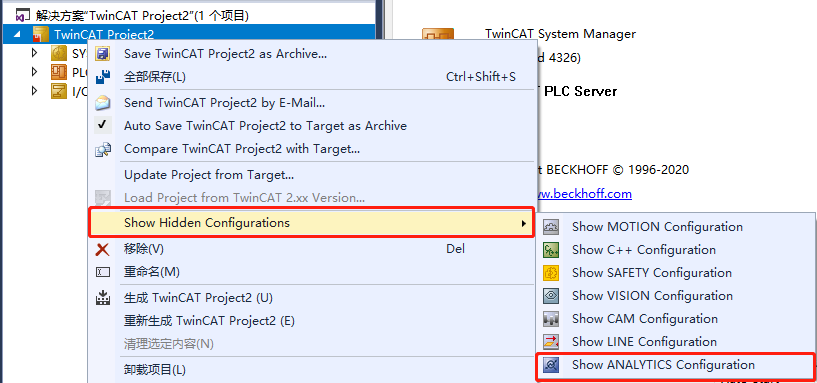
提供数据的PLC程序，最重要是把需要采集变量都加上属性{attribute 'TcAnalytics'}。

示例的PLC程序仅一个变量Main.bTest，无逻辑：

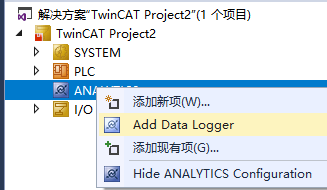
|  |
| --- |
| PROGRAM MAIN  VAR  {attribute:'TcAnalytics'}  bTest:BOOL;  END\_VAR |
| ； |

#### 配置Data Logger

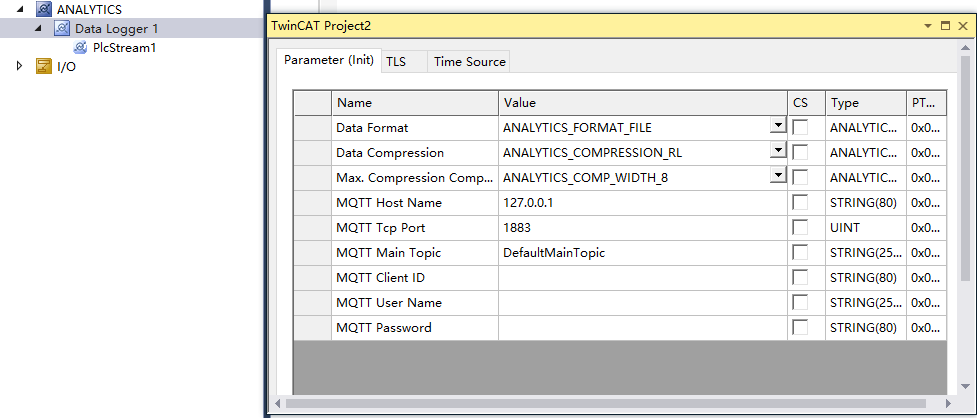
如果解决方案资源管理器中没有ANALYTICS这项，就从TwinCAT项目的右键菜单中选择Show Hidden Configurations ｜ Show ANALYTICS Configuration：



然后从ANALYTICS右键菜单选择“Add Data Logger”

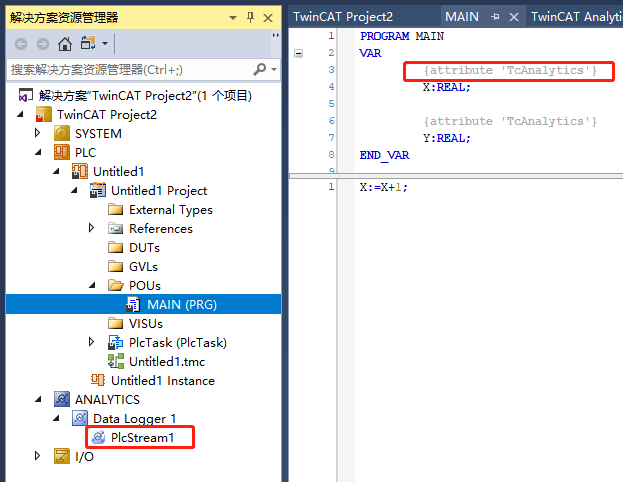


Data Logger的默认配置是这样的：



[TF3500\_TC3\_Analytics\_Logger\_en-us\_draft.pdf](file:///D:\30%20技术文档\25%20IOT\5%20TwinCAT%203%20Analytics\TF3500_TC3_Analytics_Logger_en-us_draft.pdf)第19页有每项的含义，单机本地存储时可以直接使用默认配置，打开其它两个页面可见TLS为None，Time Source为None。

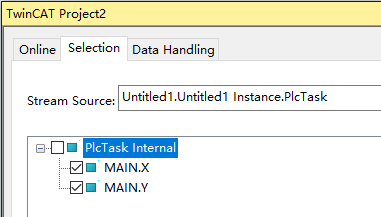
如果创建Data Logger时PLC编译过的程序中已经存在属性为{attribute:'TcAnalytics'}的变量，Data Logger下就会自动生成PLC Stream：



如果创建的时候还没有需要Logger的变量，而是之后才给PLC变量加上属性{attribute:'TcAnalytics'}，那么程序编译后，PLC Stream就会自动出现在Data Logger下。

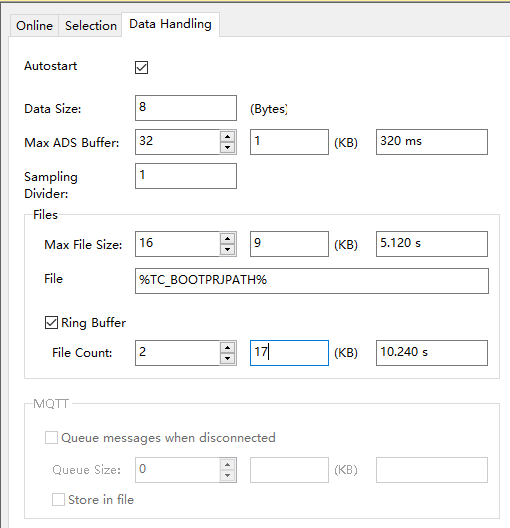
#### 选择需要采集的PLC变量

在PlcStream的Selection页面勾选变量：



只有属性为{attribute:'TcAnalytics'}的变量，才会出现在可供选择的列表中。

也可以再查看下Data Handling页面，虽然直接使用如下默认设置即可：



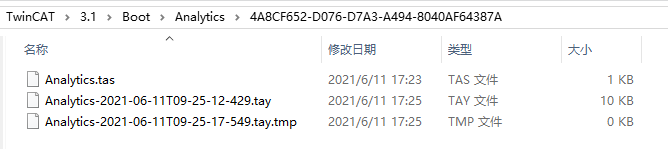
这里的File就是默认存储采集数据的文件路径，%TC\_BOOTPRJPATH%就表示“C:\TwinCAT\3.1\Boot\Analytics”

#### 激活配置并下载运行PLC程序

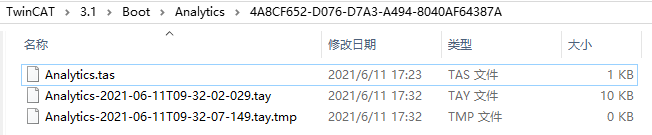


#### 在文件夹下查看数据记录的结果

根据文件夹的更新时间，可以查到最新的记录文件：



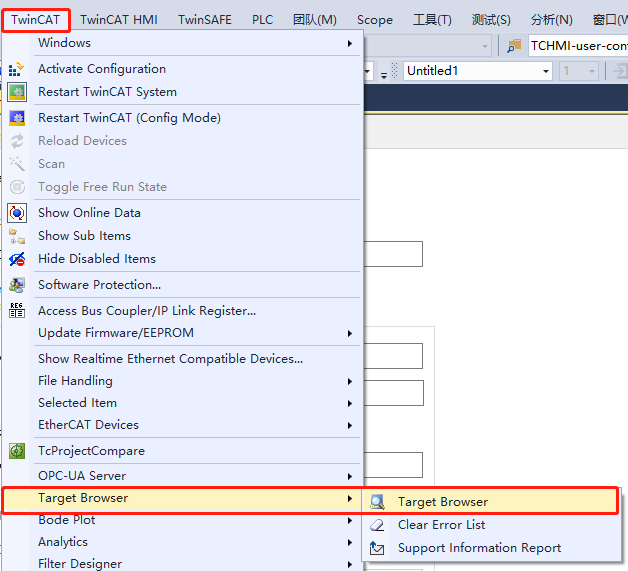
维持PLC正常运行，过几分钟再查看这个文件夹：



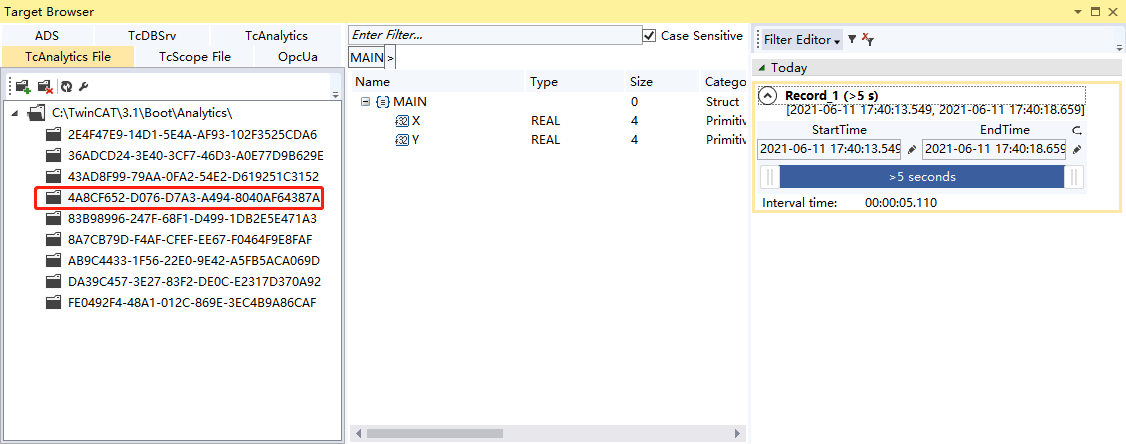
可见文件名变化了，文件大小不变。对照文件修改日期，可以推测文件名的后半部T09-32-02-029的含义是，9点32分02秒029毫秒。修改时间17:32是北京时间，而对应的格林威治时间（时区0）刚好是09:32。

文件大小不变，而两个文件的时间相差5s120ms，预计10kB数据只包含了Logger中最新5s120ms时间内的历史数据。

#### 用Target Brower观察数据记录



在TargetBrowser的TcAnalyticsFile页面找到文件夹（后4位字符387A）：

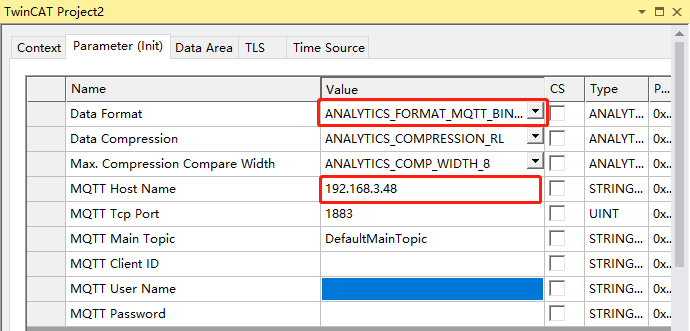


可以见到采集的变量，及记录长度，当前是00:00:05:110

### 如果通过MQTT服务器

先完成上一节“如果单机本地直接采集”的第（1）至（4）步。

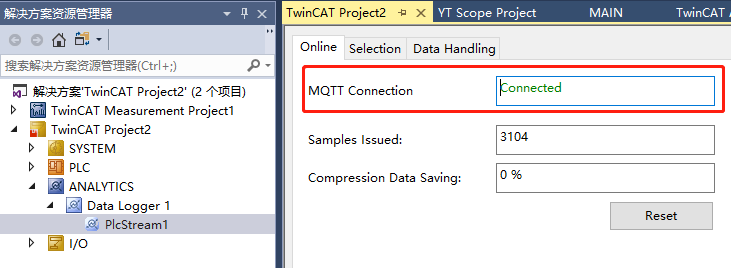
然后修改第（2）步中Data Logger的参数设置。如果通过MQTT服务器，第一项DataFormat就应选择Binary，而MQTT Host Name处要填写服务器的IP。



对于最简单的匿名访问，直接使用1883端口，Topic可以修改也可以默认，ClientID可以填或者不填。

#### 查看PlcStream的状态

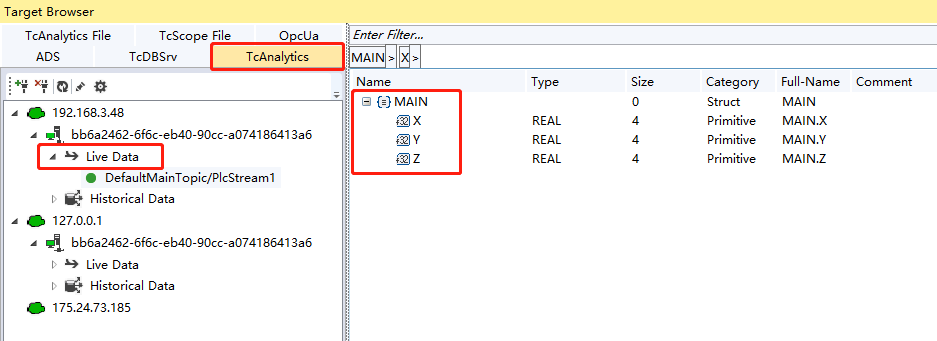
如果一切正常，控制器激活配置并且PLC启动运行后，PlcStream的Online页的MQTT Connection状态应显示为“Connected”：



提示：如果不正常就会显示红色的Disconnected，这时就需要去检查控制器的授权、网络连接、服务Data Logger服务是否启动了。如果都没问题，可以尝试设置为自动启动和引导运行PLC程序，然后重启控制器。

#### 用Target Brower观察数据记录

从主菜单TwinCAT | Target Browser | Target Browser打开目标浏览器，在TcAnalytics页面找到MQTT Broker的IP，展开树形结构显示Live Data：



如果能显示IP下的Live Data中的变量与程序中定义并在Data Logger中选择的变量一致，就说明通过MQTT的数据采集（Data Logger）成功了。

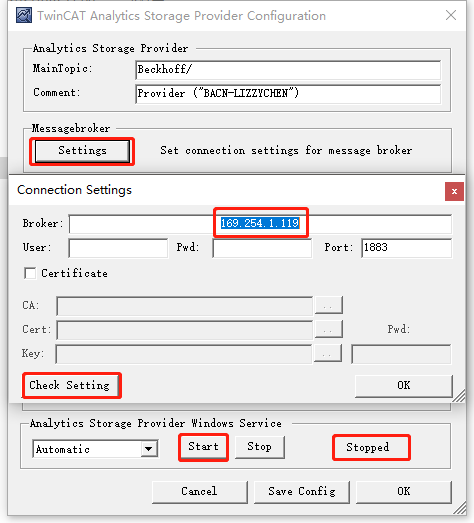
## 数据记录和存储

以下操作在配置Data Logger的同一台或者另一台安装了TC3 XAE和Storage Provider的IPC上执行。

#### 配置Storage Provider Recorder

打开目录： C:\TwinCAT\Functions\TF3520-Analytics-StorageProvider\WinService\

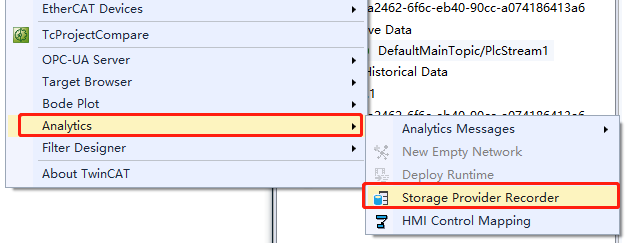
执行程序：TcAnalyticsStorageProvider\_Config.exe



设置Message Broker信息，最重要的是IP地址，匿名访问的端口都是1883。初次测试使用匿名访问即可。设置完成后点击Check Setting，显示OK就可以返回了。然后点击Start按钮，确认状态为Running。

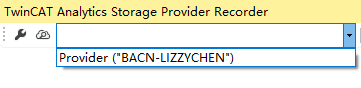
#### 打开Storage Provider Recorder

主菜单TwinCAT | Analytic | Storage Provider Recorder

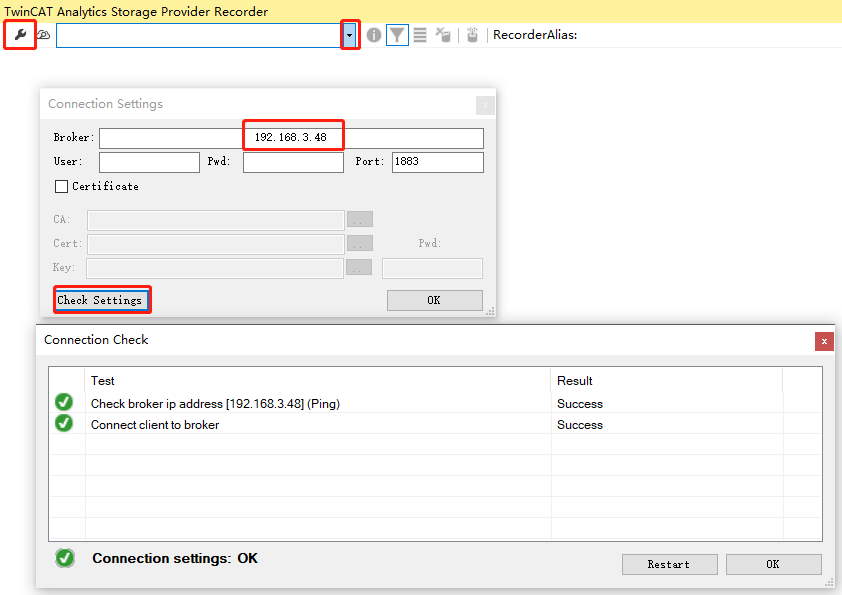


#### 选择云上的Provider

正常应该能列表显示可用的Provider：



如果列表为空，点击配置图标，检查Connection Settings：

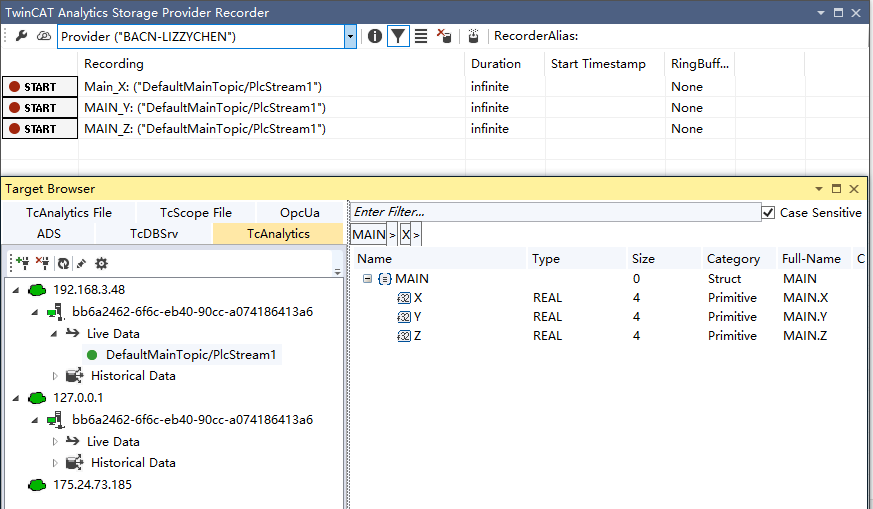


填写正确的MQTT服务器IP和端口后，点击Check Settings，正常结果应为OK。

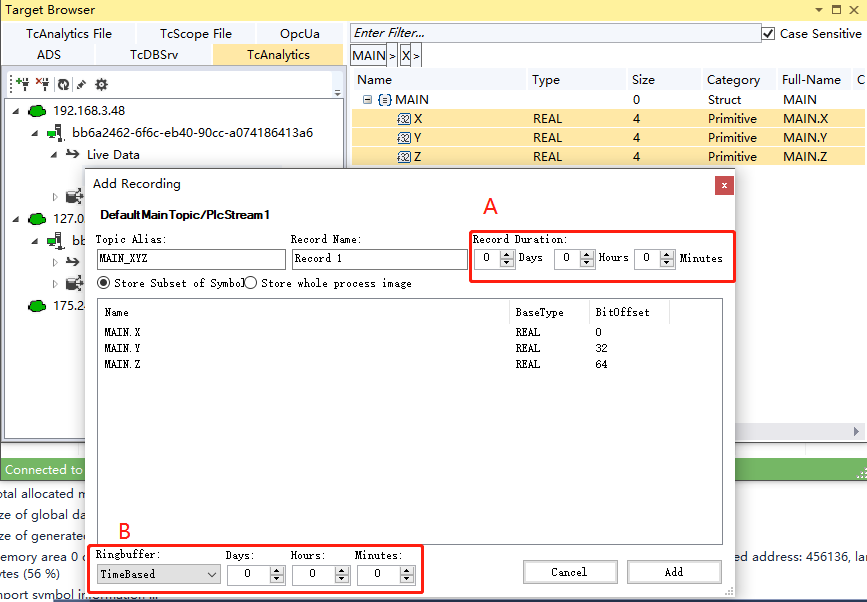
如果Broker选择的下拉框仍然为空，就需要检查本机的Provider配置和授权了。

#### 选择要存储的数据记录

从Target Browser中选择变量拖放到Storage Provider Recorder

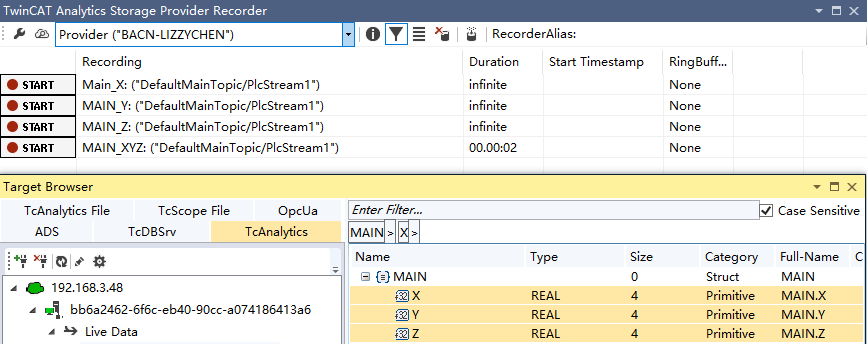


可以一次选择单个或者多个变量，然后设置Topic　Alias和记录时长：



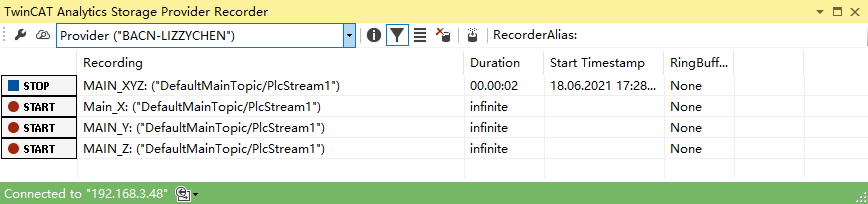
上图中A、B两处的时长设置是互相排斥的，A处指单次记录，B处指循环记录，设置A处则B处清零，反之亦然。

例如在A处设置为2分钟，结果Storage Provider Recorder中的记录项就增加了1项“Main\_XYZ”，如图所示：



#### 启动数据记录

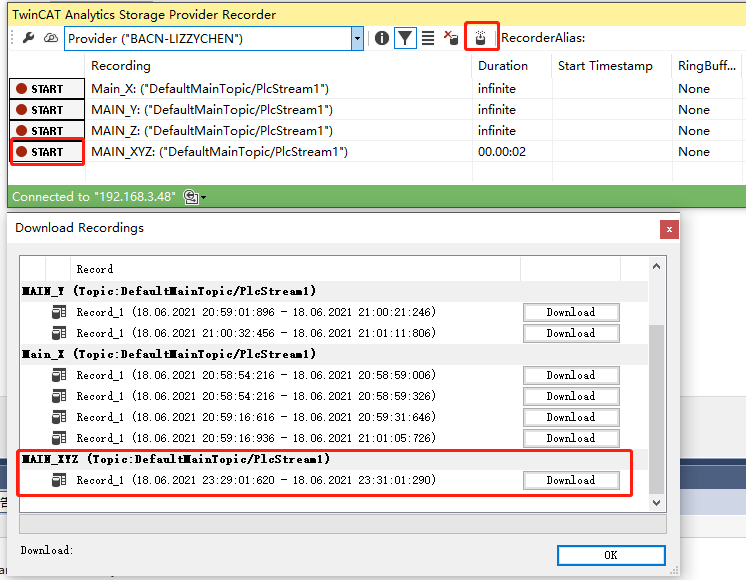
点击指定Topic前面的START按钮：



可见MAIN\_XYZ前的按钮稍后就显示为蓝色的STOP，这表示当前正在记录，可以手动停止。

#### 下载记录数据

2分钟后，MAIN\_XYZ前面的图标又显示为红色的START，表示记录已经停止，可以手动启动。每记录一次，就会生成一个文件。点击界面上方的保存按钮，选择刚才记录的MAIN\_XYZ：



点击Download，选择存储路径即可。

#### 查看数据记录的文件

到指定路径查看刚才记录的文件：



上图中.tay文件大小234kB，就是MAIN\_XYZ中3个Real型变量在2分钟内的历史记录。因为是Binary格式，可以计算文件有效占用空间。

PLC默认周期10ms，2分钟内PLC运行了12000次，每次记录3个Real数据即12字节。所以纯数据的Size为144kB。

3\*4\*（2\*60\*1000/10）=12\*12000=144,000 Byte

考虑到12000行的Timestamp，每个64位即8字节，总计96,000 Byte

数字和时间的总占用字节数为：144,000+96,000＝240,000 Byte

而计算机的1kB实际是1024字节，所以该数据记录不论保存为什么格式的文件，不应小于240,000 / 1024 = 234.375 kB。

对照截图中的.tay文件大小234kB，可见这是一个纯二进制文件，没有插入任何标记。

## Analytics 数据后处理

.tay 文件通过ScopeView 进行采集，然后转成其他格式文件

.tay 文件直接通过Tc3ScopeExportTool.exe进行处理

注意: .tas文件不能删除否则文件无法使用

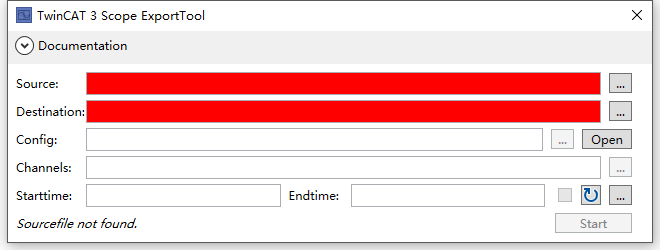
### 方法一：用TC3 Scope导出

#### 执行文件*TC3ScopeExportTool.exe*

*C:\TwinCAT\Functions\TF3300-Scope-Server\TC3ScopeExportTool.exe*

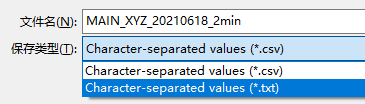
或

*C:\TwinCAT\Functions\TE130X-Scope-View\TC3ScopeExportTool.exe*

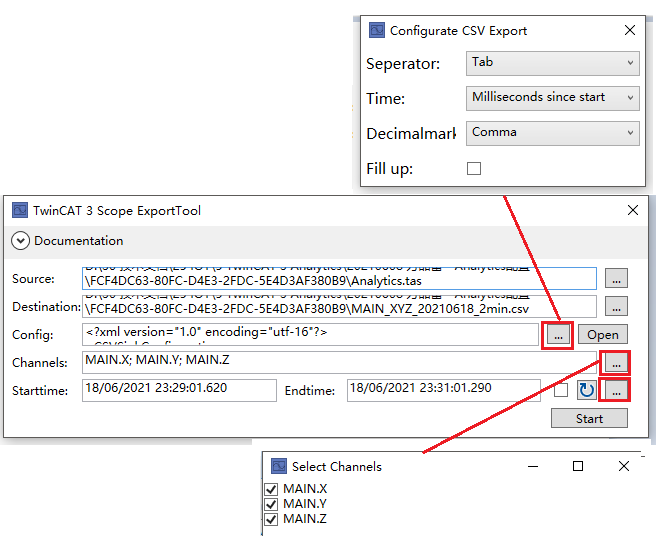


#### 选择Source和Destination

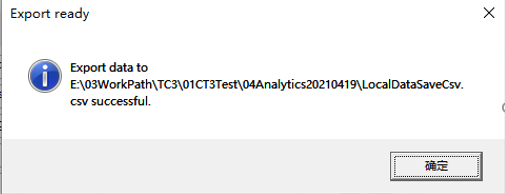
即源文件.tay和目标文件，目标文件可以是.csv或者.txt：



#### 设置文件格式、数据通道和时间



然后点击Start，正常就应该出现导出成功的提示：



### 方法二：用TC3 Scope View

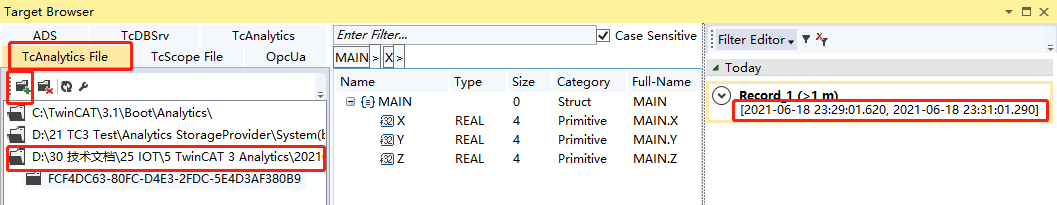
#### 确认安装TC XAE的电脑上有TE1300授权

从.tay文件中读取历史数据显示成曲线，需要TwinCAT Scope Professional授权。

如果没有，可生成Trial License，最多可以显示5个变量。

#### 新建Messurement项目

#### 打开Target Browser，拖放变量

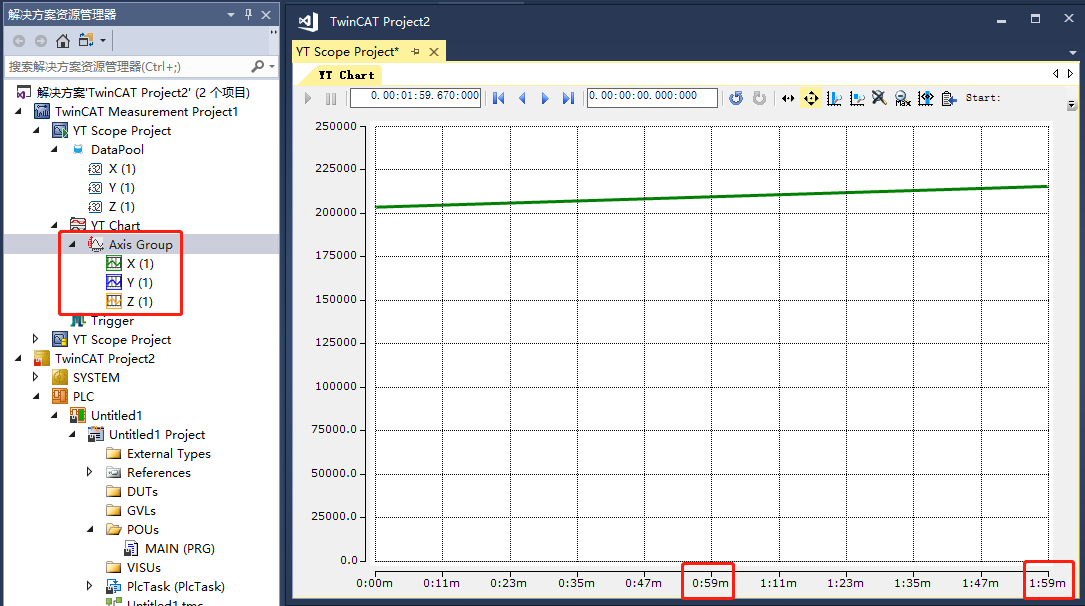


可以一次选中多个变量拖放到Scope Project。

#### 开始记录

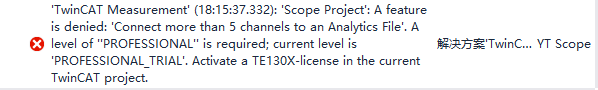
点击Scope开始记录的图标， 

就会从离线文件.tay中装截数据，就象现场从PLC中记录的数据一样：



上图可见，.tay文件记录了2分钟的数据，而Scope中显示出来的曲线就是2min。

如果变量数目超过5，Scope Project就会报错，提示需要TwinCAT Scope Professional的正式授权。



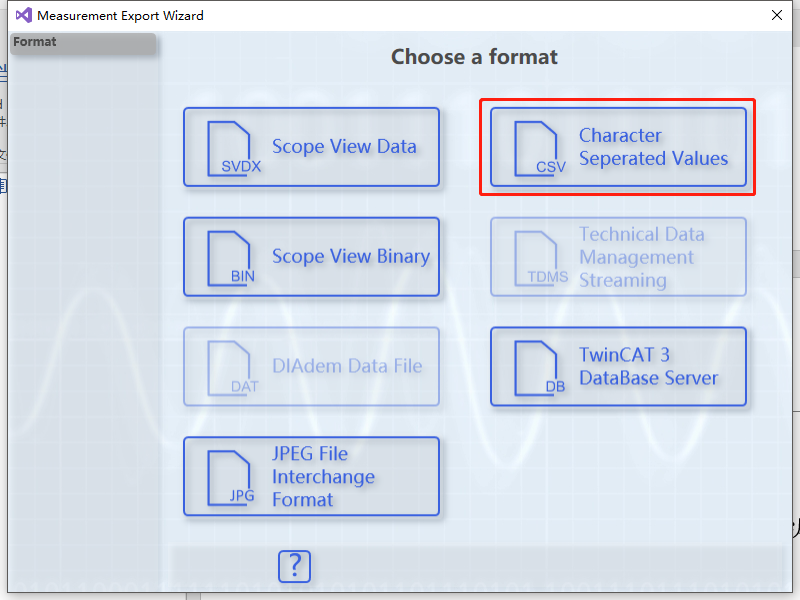
#### Scope中的其它分析功能

.tay离线文件数据装载进Scope后就可以和在线监视的变量一样，实现Scope功能中的所有分析了。

#### 从Scope导出到.csv

这是Scope标准功能，不再详述。

从主菜单Scope | Export 进入导出模板：



然后一步步按提示操作即可。

# PLC代码方式：TwinCATAnalyticsFileSaveLocal

## 准备工作

通讯双方都需要授权：TF6701 | TC3 IoT Communication

BACN库文件安装：tc3\_5gutilities.library，

路径：\AnalyticsIotMQTT\TcMLFrameWorkServer\TcMLFrameWorkServer

\MLFrameWorkServer\\_Libraries\bacn\tc3\_utilities\2.0.0.3

安装包：无

控制器：CX5130，WES7，TC3.1.4024.15(XAR)

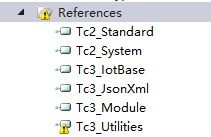
编程PC：笔记本，Win10 64位，TC3.1.4024.15，VS2017, TcXaeShell

## 项目：TcMLFrameWork

倍福中国工程师创建了这个示例项目，封装了几个关键的FB，FB\_MQTTClient用于在发送端打包结构体到Json格式送到MQTT，FB\_MQTTServer在接收端从MQTT接收Json数据再解包成结构体，FB\_StructDataSave用于存储.csv文件。需要记录的数据首先要放在结构体ST\_UserData中。

示例项目中只有一套PLC程序，通过全局变量.bClient切换角色。

### 引用库：References



​说明：上图中TC3\_Utilities是新增的库，向倍福工程师索取Demo程序，找到这个库文件后还需要单独安装。

### 自定义结构体：ST\_UserData

|  |
| --- |
| TYPE ST\_UserData :  STRUCT  iRowIndex : UDINT;  strDateTime : STRING;  iSensorValue1 : INT;  iSensorValue2 : BOOL;  iSensorValue3 : REAL;  iSensorValue4 : REAL;  iSensorValue5 : REAL;  END\_STRUCT  END\_TYPE |

### 自定义功能块：FBs

#### FB\_MQTTClient

|  |
| --- |
| FUNCTION\_BLOCK FB\_MQTTClient  VAR\_INPUT  sClientId : STRING(255):='MyTcMqttClient'; // default is generated during initialization  sTopicPub : STRING(255) := 'topicTest01'; //baidu  sHostName : STRING(255) := '175.24.73.185'; // default is local host  nHostPort : UINT := 1883; // default is 1883  sTopicPrefix : STRING(255); // topic prefix for pub and sub of this client (handled internally)  nKeepAlive : UINT := 60; // in seconds  stPubData : ST\_UserData;  bAllowedPublish : BOOL:=TRUE;  //sUserName : STRING(255); // optional parameter  //sUserPassword : STRING(255); // optional parameter  //stWill : ST\_IotMqttWill; // optional parameter  //stTLS : ST\_IotMqttTls; // optional parameter  //ipMessageFiFo : I\_IotMqttMessageFiFo; // if received messages should be queued during call of Execute()  END\_VAR  VAR\_OUTPUT  bError:BOOL;  bConnected:BOOL;  END\_VAR |

#### FB\_MQTTServer

|  |
| --- |
| FUNCTION\_BLOCK FB\_MQTTServer  VAR\_INPUT  sClientId : STRING(255):='MyTcMqttClient2'; // default is generated during initialization  sTopicSub : STRING(255) := 'topicTest01'; //baidu  sHostName : STRING(255) := '175.24.73.185'; // default is local host  nHostPort : UINT := 1883; // default is 1883  sTopicPrefix : STRING(255); // topic prefix for pub and sub of this client (handled internally)  nKeepAlive : UINT := 60; // in seconds  END\_VAR  VAR\_OUTPUT  bError :BOOL;  bConnected :BOOL;  stSubData : ST\_UserData;  bLoggerTriger : BOOL;  END\_VAR |

#### FB\_ StructDataSave

|  |
| --- |
| FUNCTION\_BLOCK FB\_StructDataSave  VAR\_INPUT  sLoggerPath : T\_MaxString := 'C:\Logger';  bLoggerTriger : BOOL;  aColumnTitle :ARRAY　[1..CSV\_MAX\_ITEMS] OF STRING(CSV\_ITEM\_SIZE);  aColumnValue : ST\_UserData;  iFileID :DINT;  END\_VAR  VAR\_OUTPUT  END\_VAR |

### 全局变量：GVL\_Logger

|  |
| --- |
| VAR\_GLOBAL  g\_stSubData : ARRAY[1..2] OF ST\_UserData;  g\_aColumnTitle : ARRAY [1..CSV\_MAX\_ITEMS] OF STRING(CSV\_ITEM\_SIZE);  END\_VAR  VAR\_GLOBAL CONSTANT  CSV\_MAX\_ITEMS : UDINT := 8; //Total Items  CSV\_ITEM\_SIZE : UDINT := 30; //max byte per Item  END\_VAR |

### 全局变量：GVL\_test

|  |
| --- |
| VAR\_GLOBAL  bClient:BOOL:=FALSE;  tName:STRING:='175.24.73.185'; //'localhost' OR '175.24.73.185'  END\_VAR |

### 全局变量：GVL

|  |
| --- |
| VAR\_GLOBAL  g\_stPubData : ST\_UserData;  END\_VAR |

### 程序：Main

|  |
| --- |
| PROGRAM PRG\_Logger  VAR  END\_VAR |
| IF GVL\_test.bClient THEN  PRG\_Logger();  ELSE  PRG\_Client();  END\_IF |

### 程序：PRG\_Logger

|  |
| --- |
| PROGRAM PRG\_Logger  VAR  fbMQTTServer :ARRAY[1..2] OF FB\_MQTTServer;  fbStructDataSave :ARRAY[1..2] OF FB\_StructDataSave;  END\_VAR |
| fbMQTTServer[1](  sClientId:= 'MyTcMqttClient11',  sTopicSub:= 'topicTest01',  sHostName:= '175.24.73.185',  nHostPort:= 1883,  sTopicPrefix:= ,  nKeepAlive:= ,  bError=> ,  bConnected=> ,  stSubData=>g\_stSubData[1] ,  bLoggerTriger=> );  fbMQTTServer[2](  sClientId:= 'MyTcMqttClient22',  sTopicSub:= 'topicTest02',  sHostName:= '175.24.73.185',  nHostPort:= 1883,  sTopicPrefix:= ,  nKeepAlive:= ,  bError=> ,  bConnected=> ,  stSubData=>g\_stSubData[2] ,  bLoggerTriger=> );    g\_aColumnTitle[1] := 'FileID';  g\_aColumnTitle[2] := 'RowIndex';  g\_aColumnTitle[3] := 'DateTime';  g\_aColumnTitle[4] := 'SensorValue1';  g\_aColumnTitle[5] := 'SensorValue2';  g\_aColumnTitle[6] := 'SensorValue3';  g\_aColumnTitle[7] := 'SensorValue4';  g\_aColumnTitle[8] := 'SensorValue5';      fbStructDataSave[1](  sLoggerPath:='C:\Logger2\Logger11' ,  bLoggerTriger:=fbMQTTServer[1].bLoggerTriger ,  aColumnTitle:= g\_aColumnTitle,  aColumnValue:=g\_stSubData[1] ,  iFileID:= 1);  fbStructDataSave[2](  sLoggerPath:='C:\Logger2\logger22' ,  bLoggerTriger:=fbMQTTServer[2].bLoggerTriger ,  aColumnTitle:= g\_aColumnTitle,  aColumnValue:=g\_stSubData[2] ,  iFileID:= 2); |

### 程序：PRG\_Client

|  |
| --- |
| PROGRAM MAIN  VAR  fbMQTTClient :ARRAY[1..2] OF FB\_MQTTClient;  END\_VAR |
| g\_stPubData.iRowIndex := g\_stPubData.iRowIndex + 1;  g\_stPubData.iSensorValue5 := g\_stPubData.iSensorValue5 + 1;  fbMQTTClient[1](  sClientId:= 'MyTcMqttClient1', // Each machine is unique  sTopicPub:= 'topicTest01', // Send and receive must be same  sHostName:= '175.24.73.185',  nHostPort:= 1883,  sTopicPrefix:= ,  nKeepAlive:= ,  stPubData:=g\_stPubData ,  bAllowedPublish:= ,  bError=> ,  bConnected=> );    fbMQTTClient[2](  sClientId:= 'MyTcMqttClient2', // Each machine is unique  sTopicPub:= 'topicTest02', // Send and receive must be same  sHostName:= '175.24.73.185',  nHostPort:= 1883,  sTopicPrefix:= ,  nKeepAlive:= ,  stPubData:=g\_stPubData ,  bAllowedPublish:= ,  bError=> ,  bConnected=> ); |

## 执行结果

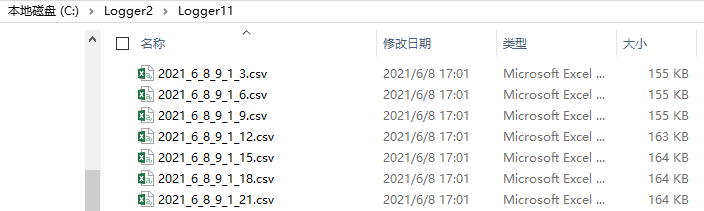
### 生成和发送数据

在PRG\_Client中发布数据至Message Broker。

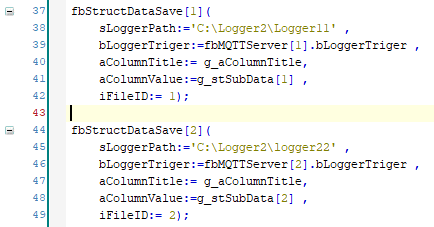
需要记录的数据放在结构体User Data中。

### 接收数据

在PRG\_Logger中接收和保存数据，来自Message Broker，保存到本地csv文件。



这个路径是在程序PRG\_Logger中定义的：



**上海（中国区总部）**

中国上海市静安区汶水路299弄9-10号（市北智汇园）

电话: 021-66312666 传真: 021-66315696 邮编：200072

**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 传真: 010-82200039 邮编：100035

**广州分公司**

广州市天河区珠江新城珠江东路16号高德置地G2603室

电话: 020-38010300/1/2 传真: 020-38010303 邮编：510623

**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 传真: 028-86202582 邮编：610016

|  |  |
| --- | --- |
| 请用微信扫描二维码  通过公众号与技术支持交流 |  |
| 倍福中文官网：  https://www.beckhoff.com.cn/ |
| 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 |