=================================================[2016.08.18]

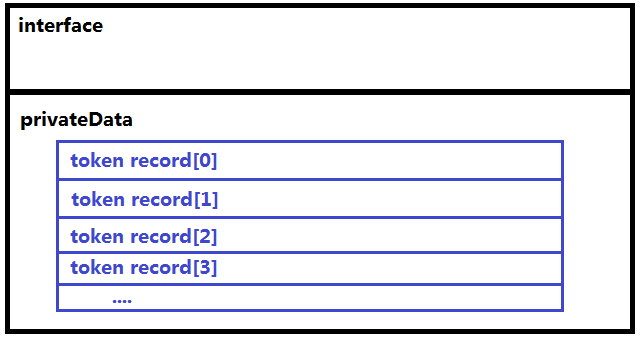
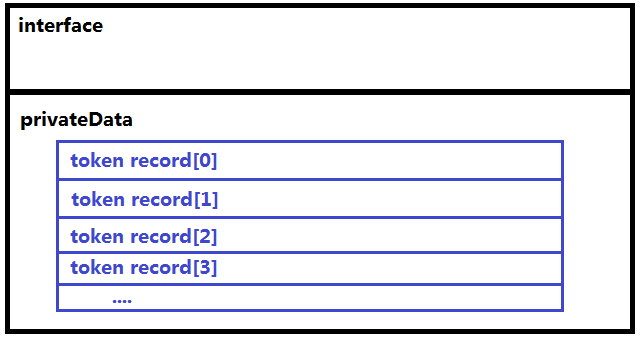
-----------V0.2--PLC的详细规划---------

零、PLC程序的描述

1. PLC内的程序负责管理（新建、修改、查找、删除等）一组令牌插座；当相关app与PLC联系并提供令牌插头信息时，负责检验令牌插头的有效性；当某个令牌有效后，负责判断app传递来的心跳信号，心跳正常则保持令牌激活，心跳停止则取消令牌激活；针对多个app的同时连接的可能性，负责为每个app分配不同的I/O地址；管理（初始化、分配、回收）一组I/O地址。
2. 由于PLC不能动态分配地址，以上的令牌插座组和I/O地址组都是固定长度的array。因此，可能存在地址用完的可能性。

一、数据结构

1、两个数据块：interface数据块和privateData数据块。interface数据块的序号必须固定，必须按地址寻址，以便于app读写。privateData数据块无此要求，建议设计成FB的背景数据块。



interface数据块存放整个PLC令牌程序的交互数据。privateData数据块存放多条令牌的数据记录，即token record。

App只能对interface数据块进行读写操作。interface数据块是PLC对app开放的数据区域。

privateData数据块存放PLC的内部数据。它是PLC内部数据的存储区域。

1. privateData中包含多条token record，每个token record内部结构如下：

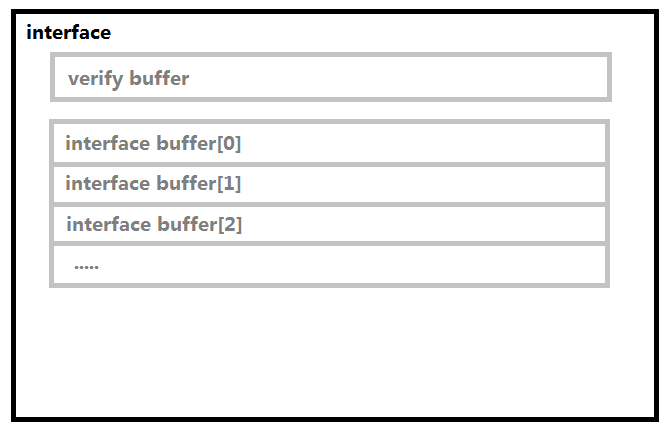


1. State byte。该字节为token record的状态。
2. App ID DWord。是app的ID，用于唯一标识该token record对应的APP。在 “配对模式”时由PLC负责录入，在“运行模式”中PLC和App都只读。
3. App Priority word。是app的优先级。在“配对模式”时由PLC负责录入，在“运行模式”中PLC和App都只读。
4. interface数据块为app提供读写的buffer。

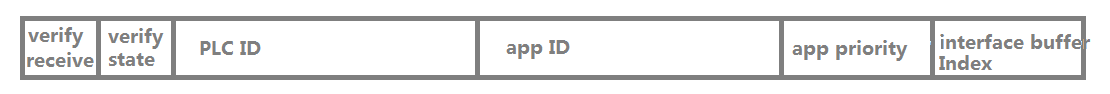
此处为解决一个难题：多个app如何同时读写PLC？因此设计了一套解决方案：app连接PLC时，需要进行两个步骤：验证、使用。

Interface数据块提供一个verify buffer。所有app连接到PLC，第一步要在verify buffer中完成验证以及地址申请工作。验证通过后，PLC会给该app提供一个独立的buffer，供app收发心跳等指令。

因此，interface数据库由以下几个部分组成：



1. Verify buffer



1. Verify receive byte。该字节接收app的指令。App写值，PLC执行并及时清零。
2. Verify state byte。PLC的状态。App只读。
3. PLC ID DWord。为了验证App中的令牌插头是否适用于本PLC。在验证过程中需要App填写PLC ID。PLC对PLC ID进行验证。App写入，PLC验证并及时清零。
4. App ID DWord。为了验证App中的令牌插头是否已经在本PLC中注册。在令牌验证过程中需要App填写App ID DWord。验证时PLC在token record中搜索相同的app ID。App写入，PLC验证并及时清零。
5. App priority。在配对时，app写入。用于告知PLC该令牌的优先级。
6. Interface buffer index word。app验证通过后，PLC给该app分配一个interface buffer。这个word的值就是其序号。给出index word后，表示本次verify过程结束。
7. interface buffer



1. receive byte。该字节接收app的指令。App写值，PLC执行并及时清零。
2. state byte。PLC的状态。App只读。
3. App ID。本Interface buffer的使用者的app ID。用于app ID的校验。
4. Timer。心跳报文的检测定时器。
5. 功能描述
6. 配对处理器

这个程序用于app和PLC配对时的令牌注册。PLC在“配对模式”时有效。

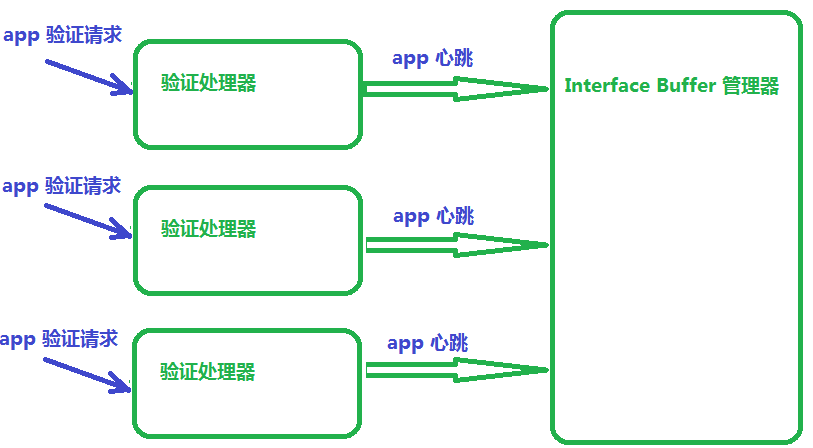
1. 基本情况：
   1. 异步执行指令，触发后异步完成。
   2. 由app指令触发，也由app指令完成
   3. 同时只能有一个app的一个令牌进行配对。
2. 触发条件：
   1. PLC处于“配对模式”。
   2. Verify receive byte接收到app发来的配对指令。
   3. 配对处理器不忙。
3. 流程：
   1. 在token record中查找是否已经注册。
   2. 若否则新增一个token record，输入app id 和app priority信息。
   3. 把本PLC的ID写入Verify buffer的PLC ID中。供app读取。
   4. 更新Verify state byte
   5. 等待Verify receive byte接收到app发来的配对完成指令。
   6. 完成后对Verify buffer清零。
4. 错误：
   1. Token已经存在
   2. Token record 已经满了
   3. 超时。长时间没有收到app的配对完成指令，取消本次配对。
5. Verify处理器

这个程序用于app第一次连接PLC时的验证。PLC在“运行模式”时该函数有效。

1. 基本情况：
   1. 异步执行指令，触发后异步完成。
   2. 由app指令触发，也由app指令完成
   3. 同时只能有一个app的一个令牌进行验证。
2. 触发条件：
   1. PLC处于“运行模式”。
   2. Verify receive byte接收到app发来的验证指令。
   3. Verify处理器不忙。
3. 流程：
   1. 检查PLC ID是否正确。
   2. 搜索app ID 和app priority对应的记录
   3. 更新Verify state byte
   4. 分配一个interface buffer index
   5. 写入该interface buffer的App ID
   6. 更新该interface buffer的state
   7. 等待Verify receive byte接收到app发来的验证完成指令。
   8. 完成后对Verify buffer清零。
4. 错误：
   1. Interface buffer已经满了
   2. PLC ID不正确，该令牌不适用于本PLC
   3. App ID和app priority未找到，该令牌未注册
   4. Verify超时。长时间没有收到app发来的验证完成指令。取消本次验证，收回分配的interface buffer
5. Interface buffer管理器

这个程序用于处理Interface buffer。PLC在“运行模式”时该函数有效。

1. 基本情况：
   1. PLC在“运行模式”时常执行指令。
2. 流程：
   1. 检查每个interface buffer的receive byte。可以收到心跳指令，也可以收到app的“拔出令牌”指令。
   2. 心跳报文定时器
   3. 收到拔出令牌指令后，free本interface buffer的占用。
3. 错误：
   1. 长时间未收到心跳，自动free本interface buffer的占用。
4. 简单流程



=================================================[2016.08.02]

-----------V0.2--规划---------

“令牌”

一、静态设计

1、每个PLC具备一个独一无二的ID，ID由Get\_IM\_Data指令获得。

2、手机上运行的令牌app也具备一个独一无二的ID。ID根据MAC生成。

3、每台PLC可以包含多个“令牌插座”。由独立的令牌管理App管理（添加、删除）这些“令牌插座”。

4、令牌app可以包含多个“令牌插头”。通过令牌app，可以新建、修改、配对“令牌插头”。

5、令牌采用某种加密算法，根据PLC的ID和app的ID产生一个密码。PLC依据该密码判断令牌是否有效。

6、PLC有“配对模式”和“运行模式”。PLC系统通过某些特殊手段（打开钥匙开关、HMI取得管理员等级）手动进入“配对模式"。该模式可以注册“令牌插座”。

二、动态设计

1、app新建“令牌插头”。令牌app上，手动新建一个令牌，设置其优先级、PLC的IP地址、SLOT、RACK等参数。令牌app连接到一台处于“配对状态”的PLC，就可以进行配对操作。

2、配对操作。令牌app读取PLC的ID，与自身的ID进行混合运算，产生密码。app和PLC都记录该密码。PLC还会记录该令牌的优先级。

3、PLC运行模式中，令牌app激活了某个令牌后，会向对应的PLC写入密码。PLC收到密码后，检验其有效性。若有效，则激活“令牌插座”

4、激活过程中，令牌app不断修改PLC的某一个寄存器，作为“心跳信号”。PLC只有持续监测到来自令牌app的心跳信号，才能保持对应的“令牌插座”持续激活。

5、

=================================================[2016.07.31]

-----------V0.1--规划---------

“app令牌”

一、令牌设计

1、每个PLC具备一个独一无二的ID，ID由Get\_IM\_Data指令获得。

2、每台手机也具备一个独一无二的ID。ID根据MAC生成。

3、令牌实际上是PLC的ID和手机ID的混合运算得到的一个数值。

一、应用场景

以下1～4点按时间顺序，简述了注册“app令牌”的过程；以下5～10点按时间顺序，简述了使用“app令牌”的过程；11开始是对系统特点的描述。

1、PLC系统通过某些特殊手段（打开钥匙开关、HMI取得管理员等级）手动进入"注册模式“。

2、安装有app的手机，与PLC连接到同一个局域网。在app上，手动新建一个“app令牌”。