

0307 【万泉河】论 PLC 算法之不可分享

上一篇文章《0302 【万泉河】PLC 编程，耦合比算法重要》发表后，反馈的效果好了点。没了那么多为了喷而喷的喷子，清净多了。

那么，后续，我会针对耦合和算法两个方面分别会再发表一些观点文章。都是针对 PLC 领域的编程方面的基础理论知识。很浅显，没难度。但很可惜，大多又没有在其他场合有人跟你讲这些。所以，如果有雷同，那只是巧合。而如果最终又被认定是我个人首先提出的观点，那么恭喜你，又快人一步从我提供的这个渠道获得了最新的编程思想。这个行业有一些顽固派，逻辑很奇怪，他会一方面反对你的观点，另一方面又要各种角度试图证明你的方法不是首创的，是他们早就掌握的。掌握了不公开发表怎么证明你掌握过？

直到前几天，有一位同业评论一段话：

有个定律，自己出生之前的技术都是理所当然，自己经历的技术都是革命性的，40 岁之后的技术都是骗人的。

令我豁然开朗，对他们有了些稍微的同理心。原谅他们，权当是可怜可怜他们吧！

他们已经老了，已经没了学习能力，但还有好强心，品德底线再低一点，很容易就这样了。当然，有的还没我年龄大，但可怕的是一眼可以看到底的到死或者到退休也不可能再拿出像样的拿的出手的成就来了，然后又爱虚荣，需要光环，就自然生出来些龌龊心。

我写文章都是给尚有学习能力的人看的，所以如果没了学习心，还真别跟着来趟这条技术线的水了，留下的只是被同行耻笑的笑柄。

不过，我也还要给大家灌输一个读文章的方法：我写一篇文章，几千字，整理的是我的思想总结，或者技术方法。工业技术领域，所有的方法都是可简单验证的，而我们从事的 PLC 行业，我提出的这些基础的编程思想方法，甚至都不需要硬件，就在电脑上用编程软件，顶多加上模拟软件，分分钟可以验证，所以就根本不可能造假，也无从打假。有疑问，自己稍微勤快一点，电脑开起来，软件支起来，很短时间就可以验证，没必要来跟我争论质疑。

还好比，我曾经分享过 80 系列的多个简单例子程序，包括 80 模拟量和 80 双位开关，在多个 PLC 平台用同等方法实现。目的就是要展示 PLC 程序可以做到简单简洁，而且可以多平台一致性。我本意是想向大家传授这就是最简单最简练的标准答案的，除非有人可以提出更简练的方法来和我 PK。当然我也欢迎这种 PK。

但没想到，有人竟然把自己当审判员了，嘲笑我程序太简单，在他眼里不够好。我就直接告诉他，实现同样的功能目的，越简单的程序才是好程序。如果没有这样的认知，那就是自己的价值观错了。

80 系列的程序请在公众号后台回复 800 得到。里面有我做的十几套各品牌的例子程序。以前放在一个公开的 QQ 群的，太容易满员了，后来就放弃了。

一定要记住：简单的程序才是好程序！

终于进入正题

我为什么主张算法不可分享呢？主要原因是我认为分享算法没意义。

首先，PLC 行业，通用的算法都太简单，无非一些循环加跳转之类的语法，没有人需要通过读别人文章来学习简单语法。如果他需要，那就说明他根本还没有入门，也当然不可能指望读了一段分享代码就学会了。而至于一些复杂工艺的算法，通常同样从事的人少，读者不多。发布了，读者也根本读不懂。同时作者自己还会有一些保密需求，就更不适合分享了。

公众号里推荐了很多这样的文章，比如：

《逆天脑洞！SCL 版猴子排序算法》

《GX works 2 简单工程 FB 块 --D 寄存器（16 位）查找为 1 的最低位/ 按位设置 1/ 按位设置 0》

这样的文章，作者在那儿费劲思想把算法表达明白，然而读者匆匆而过，根本不会有人留意去读懂每一句指令，也更不会有人去亲自演练操作。

要想有意义，做成一个完整的功能，封装好，提供功能说明。需要者拿到了简单验证功能可用，就可以收藏成为自己的知识库。而至于实现的算法原理，有兴趣的时候，还可以再慢慢研究。

这对复杂工艺也仍然适用。

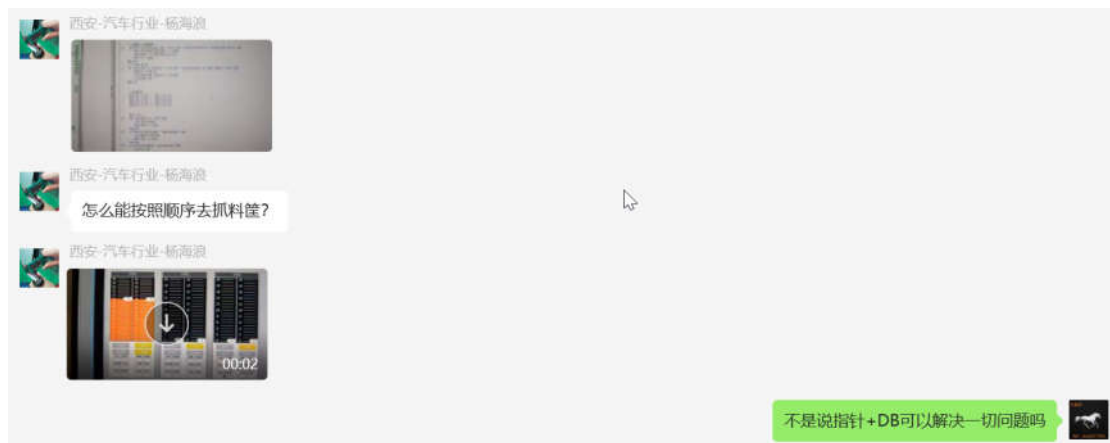
完全可以封装成完整的功能包，免费或者付费的方式提供给同样需求的人。

所以，封装作为产品，以及封装的技能，才是更值得分享的。

再举一个例子，这个行业总有一些闲人，为了练手，会在 PLC+HMI 的环境中实现一些游戏功能。比如扫雷，俄罗斯方块，纸牌游戏等等。然后也会大方的把源代码全盘共享。我同样认为这是无意义的。甚至做的过程意义也不大。如果确实要练手，确实要做，我认为更有意义的是去开源社区找来这个项目的源代码，不管它原来的编程环境是什么，你想法给移植到自动化的平台实现。

实现了以后，可以重点分享的是这个移植的过程。那么，读者通过了解你现在实现的过程，比如你实现了扫雷游戏，他或许可以如法炮制实现纸牌游戏的移植，以及更多有意义的工作中用得的功能的移植。而你self更是先行一步，掌握了全套的技能，未来工作中打下了技能储备。

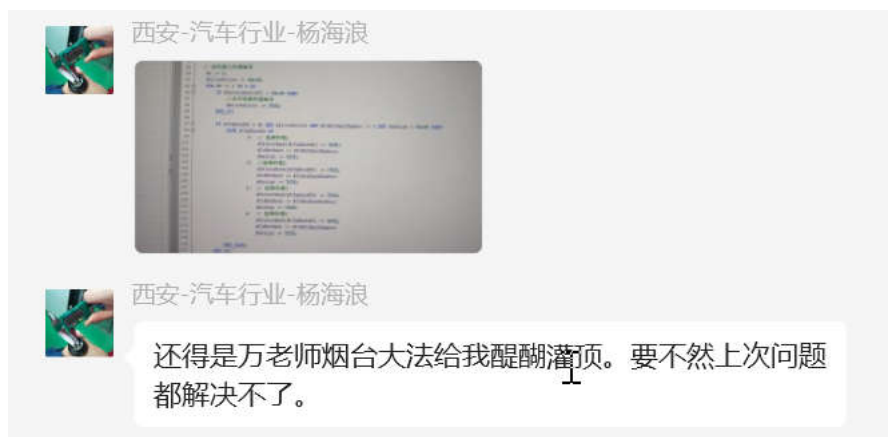
前段时间，有一个比较骄傲的小哥，曾经对自己编程水平颇为得意的，在群里提了自己搞不定的一个编程算法的问题，我其实不了解他的行业工艺，根本不懂他的问题。



但盲猜他遇到的难点，就指点他去读我一篇很早前发表过的文章：

《0707 【万泉河】更具程序员思维的 U8 轮巡题目答案》

然后很快，他就反馈，搞定了！在以后的时间里，还反复多次提及：



```

87
88 // 给机器人料框编号
89 #P := 0;
90 #AllowSelcet := FALSE;
91 FOR #P := 1 TO 4 DO
92     IF #SelectBank[#P] = FALSE THEN
93         // 允许选择料框编号
94         #AllowSelcet := TRUE;
95     END_IF;
96
97 IF #CfgBankNO = #P AND #AllowSelcet AND #ToRbtBankNumber <> 0 AND #Assign = FALSE THEN
98     CASE #CfgBankNO OF
99         1: // 选择料框1
100             #SelectBank[#CfgBankNO] := TRUE;
101             #ToRbtBank := #ToRbtBankNumber;
102             #Assign := TRUE;
103         2: // 选择料框2
104             #SelectBank[#CfgBankNO] := TRUE;
105             #ToRbtBank := #ToRbtBankNumber;
106             #Assign := TRUE;
107         3: // 选择料框3
108             #SelectBank[#CfgBankNO] := TRUE;
109             #ToRbtBank := #ToRbtBankNumber;
110             #Assign := TRUE;
111         4: // 选择料框4
112             #SelectBank[#CfgBankNO] := TRUE;
113             #ToRbtBank := #ToRbtBankNumber;
114             #Assign := TRUE;
115     END_CASE;
116 END_IF;
117

```

而他最后实现的程序的算法，我也仍然看不懂。不知道实现了什么功能，解决了什么难题。

所以，算法不重要，心法才重要。我给他提供的是心法，算法他自己实现就可以了。

80 系列的程序请在公众号后台回复 800 得到。

请注意看到我主要在演示封装后调用的简练。有人会觉得这太简单，不实用。比如项目中还要有触摸屏或者上位机，还要有很多数据处理，就不会这么简单。其实去看我的著作，里面都有讲到。触摸屏与上位机的数据通讯，也都是可以提前封装的，封装之后在应用层面呈现出来的就是简单的罗列调用，用多少调用多少。

甚至包括 MODBUS 通讯也是。

