

0317 【万泉河】从模拟量的量纲说起。

有人会问量纲是啥，就是物理单位啦！单位只是通俗的说法，而且有歧义。而物理和力学里面都称之为量纲的，既专业且没有歧义。

在英语里面这个单词叫做 Unit。

问大家一个问题，控制系统中，通常采集上来大量的模拟量数据，会显示在触摸屏或上位机上。那么每一个信号的物理单位的量纲如何标注显示，如何实现呢？

有人会笑了。这还不简单嘛，在触摸屏上每一个数据框后面添加一个静态文本，文本内容逐个修改为相对应的单位即可，与标注这个数据的标题描述方法一样。程序修改完成，下载到触摸屏中，自然就实现了。

是的，这是每个初学者都会的做法。然而也是仅仅刚刚培训学校毕业，刚刚离开教室，屁股后面的大门都还没关严实的水平。

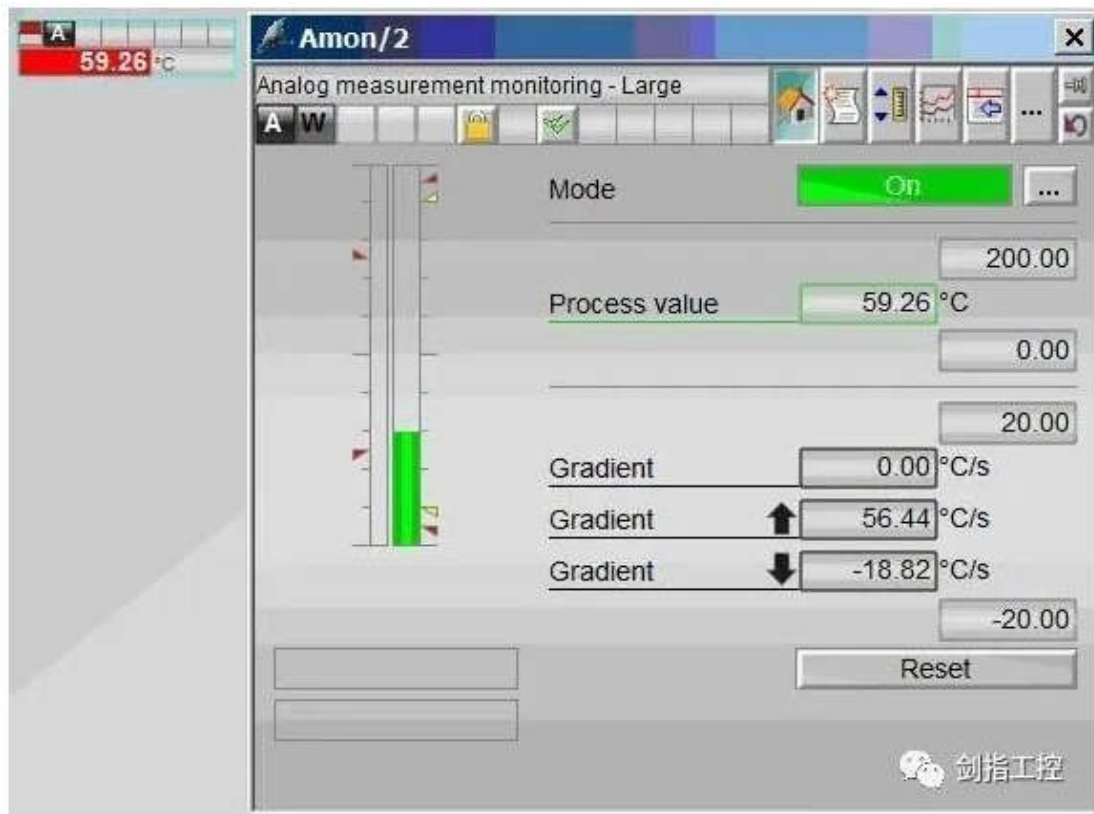
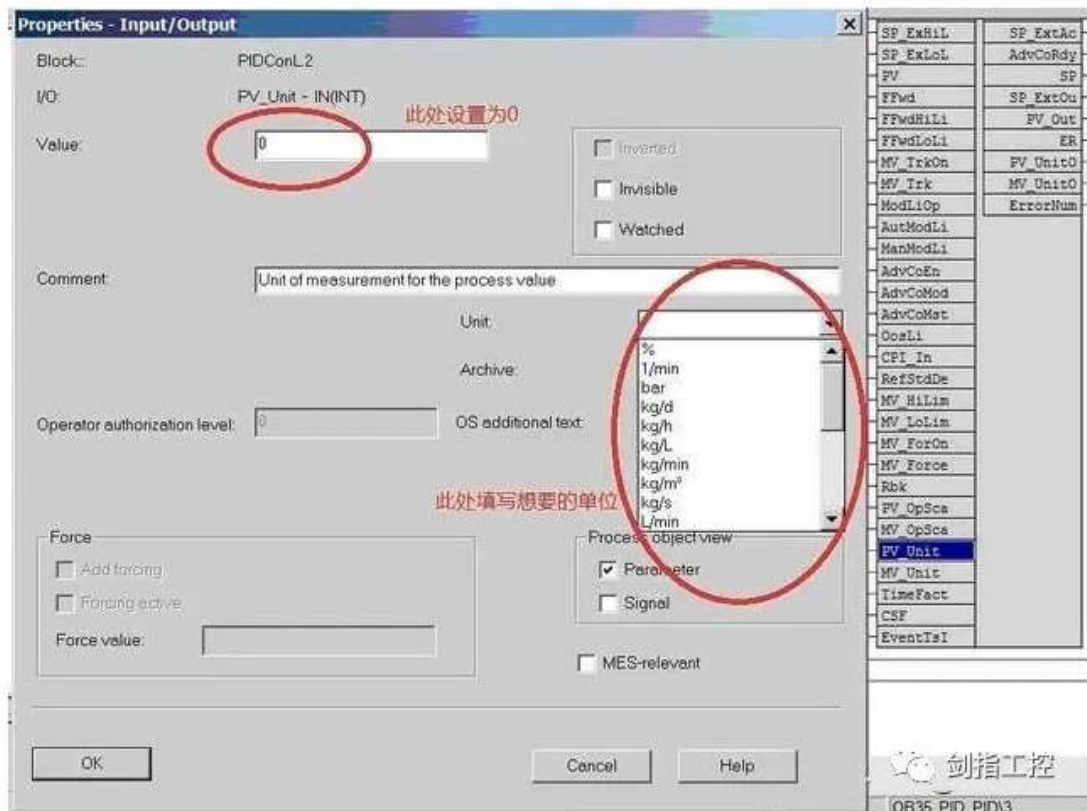
想一想你做的监控系统，模拟量数据的数量，不管是 8 个也好，80 个，800 个也好，这些数据肯定都不是一个规格，而是各种物理数据，温度，压力，流量，重量，体积，长度等等。那么每一个数据，你都需要左手翻着工艺清单，右手翻着触摸屏画面，挨个儿找过来，逐个对应，输入，修改，检查。这工作量着实不少。

而如果水平高一点，实现了弹出式窗口的管理，不管是触摸屏还是 WINCC，好像都有点麻烦了。窗口使用的同一个模板，同样的位置，量纲需要动态变换，所以还需要做些特殊处理。

然后就会质问了，除此之外难道还会有更好的方式？

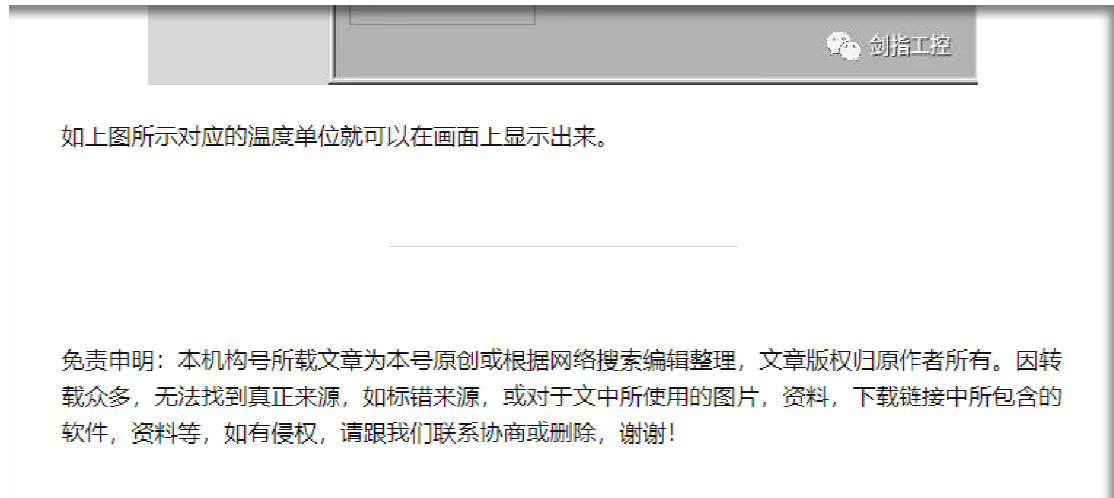
当然有。方法便是在 PLC 编程中，在调用模拟量的 FB 块的过程中录入到其管脚上。管脚数据除了物理通道地址，标定上下限值之外，再增加一个 UNIT 的字符类型的管脚即可。程序块内部甚至都可以不需要再做任何逻辑，上位机变量直接链接到这个管脚，读取其内容，并动态显示即可。

而且这都不是我一个人发明的，我只是在推广一种常识方法。让我们来看一看 PCS7 是如何做到的。



即他的 FB 有个 UNIT 的管脚，输入了量纲编码，最后在画面上实现了动态显示。

图片来自剑指工控发的网文，然而他们又备注了文章来自网络，原作者不详。剑控把抄袭的原作者的文章全都打上了他们自己的标志，然后到处分发，导致原作者的文章出处反而找不到了。



其实这里文章介绍的还是 PCS7 较新的版本，我原本可以找到自己的老版本的 PCS7 的虚拟机打开来截屏的，那时候的 UNIT 直接就是个字符串。

我拿 PCS7 来为自己背书，无非是说明，我所介绍的方法是业界的通行标准做法，并没有什么新奇之处。我做的 80 模拟量的处理程序例程，我敢有底气称之为标准答案，只不过是了解 PCS7 的这种做法。许多懂 PCS7 的朋友见到我给的例程也都觉得很自然，很正常，就是应该这个样子的。

而有一些同行，见到之后比较惊讶，只是因为你没见过 PCS7 而已。而有人倒过来嘲笑我罗列 80 行的做法很简单很幼稚。

```

CALL 模拟量_SBR19_AI_V019:AIW10. 0.0, 100.0, AI_V019_QOUT_VD2000
CALL 模拟量_SBR19_AI_V020:AIW12. 0.0, 500.0, AI_V020_QOUT_VD2004
CALL 模拟量_SBR19_AI_V021:AIW14. 0.0, 100.0, AI_V021_QOUT_VD2008
CALL 模拟量_SBR19_AI_V022:AIW16. 0.0, 100.0, AI_V022_QOUT_VD2012
CALL 模拟量_SBR19_AI_V023:AIW18. 0.0, 500.0, AI_V023_QOUT_VD2016
CALL 模拟量_SBR19_AI_V024:AIW20. 0.0, 100.0, AI_V024_QOUT_VD2020
CALL 模拟量_SBR19_AI_V025:AIW22. 0.0, 100.0, AI_V025_QOUT_VD2024
CALL 模拟量_SBR19_AI_V026:AIW24. 0.0, 500.0, AI_V026_QOUT_VD2028
CALL 模拟量_RTD_SBR2_AI_V027:AIW26. AI_V027_QOUT_VD2032
CALL 模拟量_RTD_SBR2_AI_V028:AIW28. AI_V028_QOUT_VD2036
CALL 模拟量_RTD_SBR2_AI_V029:AIW30. AI_V029_QOUT_VD2040
CALL 模拟量_RTD_SBR2_AI_V030:AIW32. AI_V030_QOUT_VD2044
CALL 模拟量_SBR19_AI_V031:AIW34. 0.0, 100.0, AI_V031_QOUT_VD2048
CALL 模拟量_SBR19_AI_V032:AIW36. 0.0, 100.0, AI_V032_QOUT_VD2052
CALL 模拟量_SBR19_AI_V033:AIW38. 0.0, 500.0, AI_V033_QOUT_VD2056
CALL 模拟量_SBR19_AI_V034:AIW40. 0.0, 100.0, AI_V034_QOUT_VD2060
CALL 模拟量_SBR19_AI_V035:AIW42. 0.0, 100.0, AI_V035_QOUT_VD2064
CALL 模拟量_SBR19_AI_V036:AIW44. 0.0, 500.0, AI_V036_QOUT_VD2068
CALL 模拟量_SBR19_AI_V037:AIW46. 0.0, 100.0, AI_V037_QOUT_VD2072
CALL 模拟量_SBR19_AI_V038:AIW48. 0.0, 100.0, AI_V038_QOUT_VD2076
CALL 模拟量_RTD_SBR2_AI_V039:AIW50. AI_V039_QOUT_VD2080
CALL 模拟量_RTD_SBR2_AI_V040:AIW52. AI_V040_QOUT_VD2084
CALL 模拟量_RTD_SBR2_AI_V041:AIW54. AI_V041_QOUT_VD2088
CALL 模拟量_RTD_SBR2_AI_V042:AIW56. AI_V042_QOUT_VD2092
CALL 模拟量_odot_SBR1_AI_V043:AIW58. 0.0, 500.0, AI_V043_QOUT_VD2096
CALL 模拟量_odot_SBR1_AI_V044:AIW60. 0.0, 100.0, AI_V044_QOUT_VD2100
CALL 模拟量_odot_SBR1_AI_V045:AIW62. 0.0, 100.0, AI_V045_QOUT_VD2104
CALL 模拟量_odot_SBR1_AI_V046:AIW64. 0.0, 500.0, AI_V046_QOUT_VD2108
CALL 模拟量_odot_SBR1_AI_V047:AIW66. 0.0, 100.0, AI_V047_QOUT_VD2112
CALL 模拟量_odot_SBR1_AI_V048:AIW68. 0.0, 100.0, AI_V048_QOUT_VD2116
CALL 模拟量_odot_SBR1_AI_V049:AIW70. 0.0, 500.0, AI_V049_QOUT_VD2120
CALL 模拟量_odot_SBR1_AI_V050:AIW72. 0.0, 100.0, AI_V050_QOUT_VD2124
CALL 模拟量_odot_SBR1_AI_V051:AIW74. 0.0, 100.0, AI_V051_QOUT_VD2128
CALL 模拟量_odot_SBR1_AI_V052:AIW76. 0.0, 500.0, AI_V052_QOUT_VD2132
CALL 模拟量_odot_SBR1_AI_V053:AIW78. 0.0, 100.0, AI_V053_QOUT_VD2136
CALL 模拟量_odot_SBR1_AI_V054:AIW80. 0.0, 100.0, AI_V054_QOUT_VD2140
CALL 模拟量_odot_SBR1_AI_V055:AIW82. 0.0, 500.0, AI_V055_QOUT_VD2144
CALL 模拟量_odot_SBR1_AI_V056:AIW84. 0.0, 100.0, AI_V056_QOUT_VD2148
CALL 模拟量_odot_SBR1_AI_V057:AIW86. 0.0, 100.0, AI_V057_QOUT_VD2152
CALL 模拟量_odot_SBR1_AI_V058:AIW88. 0.0, 500.0, AI_V058_QOUT_VD2156
CALL 模拟量_odot_SBR1_AI_V059:AIW90. 0.0, 100.0, AI_V059_QOUT_VD2160
CALL 模拟量_odot_SBR1_AI_V060:AIW92. 0.0, 100.0, AI_V060_QOUT_VD2164
CALL 模拟量_odot_SBR1_AI_V061:AIW94. 0.0, 500.0, AI_V061_QOUT_VD2168
CALL 模拟量_odot_SBR1_AI_V062:AIW96. 0.0, 100.0, AI_V062_QOUT_VD2172
CALL 模拟量_odot_SBR1_AI_V063:AIW98. 0.0, 100.0, AI_V063_QOUT_VD2176
CALL 模拟量_odot_SBR1_AI_V064:AIW100. 0.0, 500.0, AI_V064_QOUT_VD2180
CALL 模拟量_odot_SBR1_AI_V065:AIW102. 0.0, 100.0, AI_V065_QOUT_VD2184
CALL 模拟量_odot_SBR1_AI_V066:AIW104. 0.0, 500.0, AI_V066_QOUT_VD2188
CALL 模拟量_odot_SBR1_AI_V067:AIW106. 0.0, 100.0, AI_V067_QOUT_VD2192
CALL 模拟量_odot_SBR1_AI_V068:AIW108. 0.0, 100.0, AI_V068_QOUT_VD2196
CALL 模拟量_odot_SBR1_AI_V069:AIW110. 0.0, 100.0, AI_V069_QOUT_VD2200
CALL 模拟量_odot_SBR1_AI_V070:AIW112. 0.0, 500.0, AI_V070_QOUT_VD2204
CALL 模拟量_odot_SBR1_AI_V071:AIW114. 0.0, 100.0, AI_V071_QOUT_VD2208
CALL 模拟量_odot_SBR1_AI_V072:AIW116. 0.0, 100.0, AI_V072_QOUT_VD2212
CALL 模拟量_odot_SBR1_AI_V073:AIW118. 0.0, 500.0, AI_V073_QOUT_VD2216
CALL 模拟量_odot_SBR1_AI_V074:AIW120. 0.0, 100.0, AI_V074_QOUT_VD2220
CALL 模拟量_odot_SBR1_AI_V075:AIW122. 0.0, 100.0, AI_V075_QOUT_VD2224
CALL 模拟量_odot_SBR1_AI_V076:AIW124. 0.0, 500.0, AI_V076_QOUT_VD2228
CALL 模拟量_odot_SBR1_AI_V077:AIW126. 0.0, 100.0, AI_V077_QOUT_VD2232
CALL 模拟量_odot_SBR1_AI_V078:AIW128. 0.0, 100.0, AI_V078_QOUT_VD2236
CALL 模拟量_odot_SBR1_AI_V079:AIW130. 0.0, 500.0, AI_V079_QOUT_VD2240
CALL 模拟量_odot_SBR1_AI_V080:AIW132. 0.0, 100.0, AI_V080_QOUT_VD2244
CALL 模拟量_odot_SBR1_AI_V081:AIW134. 0.0, 100.0, AI_V081_QOUT_VD2248
CALL 模拟量_odot_SBR1_AI_V082:AIW136. 0.0, 500.0, AI_V082_QOUT_VD2252
CALL 模拟量_odot_SBR1_AI_V083:AIW138. 0.0, 100.0, AI_V083_QOUT_VD2256
CALL 模拟量_odot_SBR1_AI_V084:AIW140. 0.0, 100.0, AI_V084_QOUT_VD2260
CALL 模拟量_odot_SBR1_AI_V085:AIW142. 0.0, 500.0, AI_V085_QOUT_VD2264
CALL 模拟量_odot_SBR1_AI_V086:AIW144. 0.0, 100.0, AI_V086_QOUT_VD2268
CALL 模拟量_RTD_SBR2_AI_V087:AIW146. AI_V087_QOUT_VD2272
CALL 模拟量_RTD_SBR2_AI_V088:AIW148. AI_V088_QOUT_VD2276
CALL 模拟量_RTD_SBR2_AI_V089:AIW150. AI_V089_QOUT_VD2280
CALL 模拟量_RTD_SBR2_AI_V090:AIW152. AI_V090_QOUT_VD2284
CALL 模拟量_RTD_SBR2_AI_V091:AIW154. AI_V091_QOUT_VD2288
CALL 模拟量_RTD_SBR2_AI_V092:AIW156. AI_V092_QOUT_VD2292
CALL 模拟量_RTD_SBR2_AI_V093:AIW158. AI_V093_QOUT_VD2296
CALL 模拟量_RTD_SBR2_AI_V094:AIW160. AI_V094_QOUT_VD2300
CALL 模拟量_RTD_SBR2_AI_V095:AIW162. AI_V095_QOUT_VD2304
CALL 模拟量_RTD_SBR2_AI_V096:AIW164. AI_V096_QOUT_VD2308
CALL 模拟量_RTD_SBR2_AI_V097:AIW166. AI_V097_QOUT_VD2312
CALL 模拟量_RTD_SBR2_AI_V098:AIW168. AI_V098_QOUT_VD2316

```

嗯，没错，我们做程序的目的，就是要复杂问题尽量简单化，越简单越实用。而不是倒过来。

有人会抬杠说你们把这些非运算数据都放到 PLC 里面来，会增大和浪费 CPU 的系统资源。你们那是 PCS7， S7-400，你们有钱无所谓，俺们穷，用的小 PLC，这样做资源浪费。

嗯，这样的逻辑相当于你出门打个出租车，你副驾驶还是后排座全都不坐，你去站在车屁股后面推着跑，声称这样比较省油。

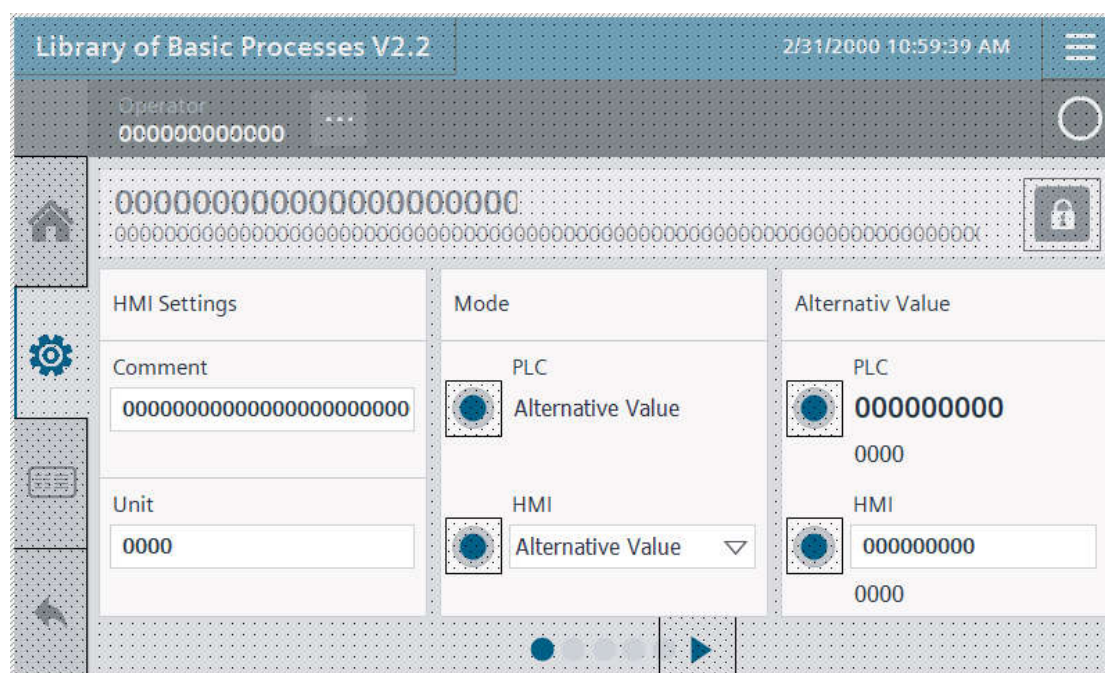
而出租车司机对这样的做法会感激你夸奖你吗？夸奖你花了同样的钱，但给他省了油吗？不会，他会骂你神经病，骂你耽误了时间，降低了效率。

其实实情是，现在的趋势是，都不仅仅是量纲这么 3-4 个字节的字符了，物理信号的位号，以及一大段的文字注释，都会放在 PLC 中实现录入，运行中 HMI 上面动态读取，节省大量的人工，提高效率。

这样的话，这些内容的录入是和数据信息是在同一个时刻实现的，不需要多次重复回来翻阅同一个工艺数据表格，提高效率同时还降低了出错的概率。甚至，从表格到 PLC 程序，可以用软件工具实现自动转换，自动生成。可以是 EXCEL 中处理，也可以用 PYTHON 等高级语言制作小工具完成。使得自动化的完成自动化设计工作成为可能。

这是很多明眼人看到我给出的标准答案之后，一眼就看懂的原因。

让我们来看一下 LBP 中这部分是怎么实现的：



LBP 的 UNIT 和 COMMENT，并没有放到 FB 的管脚上在调用时录入，而是在其内部的变量中设置了相应内容，呈现到触摸屏中。运行中从这个设定界面里人工输入，完成后运行界面中也都能正常显示正确的量纲 Unit 了。

即如果系统中有 80 个模拟量，就需要跳转 80 次页面，逐个人工输入。如果是由操作工来输入，他会骂死设计者。如果设计者自己来输入，那还不如在电脑中编程软件中输入效率高了。而且这还没考虑到系统掉电或者程序重新下载的情形，极有可能，一不小心，就会要你多次录入，给你好看。

所以，我现在倡导的学习和使用 LBP，并不是一味的照抄他，而是汲取其有营养的精华的部分，而对于不适用我们各自的行业应用习惯的部分，要在吸收的同时加以改进。

包括上面图中的 MODE，代表了通道的电信号类型的选择，也令人不太理解为什么要做在触摸屏上，我们只能承认不理解，或许有其他的行业需要这样的功能。但我们可以根据我们自己的行业的需求，修改到适应我们的模式。

针对近期对 LBP 的模拟量块的改进，结合前面提出的 80 模拟量标准答案的问题，准备举办一次公开的研讨会。暂定了一个时间，然而具体时间还要等与会人员凑齐了之后再决定。欢迎报名参加。

报名信息 and 会议内容说明，也会同时发在公众号的下一篇文章，以及朋友圈中。



腾讯会议



## 万泉河-80模拟量标准答案研讨

🆔 187 978 588

🕒 2023-03-24 (周五) 20:00-21:00

 腾讯会议

长按识别二维码参加



