

使用 EXCELVBA 进行管道试压包划分

栾德珩

中石化第十建设有限公司, 山东 青岛 266000

[摘要] 试压包划分是管道安装工程项目中的一道重点工序。划分的原则是根据管道系统的介质、压力、材质等进行划分。同一个试压包内管道等级或探伤比例可以不同, 但试验压力要相同或相近。常规试压包划分一般是参照流程图抽取互相连接的单线图进行划分。流程图纸板上想要查找到一条线, 即使是使用电子版 (PDF 格式的文件) 查找也不方便。因此流程图划分试压包容易出现漏线情况, 最后剩余未抽完的单线图需要再次进行核对。

如果不参考流程图直接使用单线图进行试压包划分, 则需要按照单线图的目录从第一条管线开始逐条进行比对。虽然有目录参考, 管线划分可以做到没有遗漏, 但是装置管网错综复杂, 分支连接跳跃频繁。要把所有相连的管线整合到一起也并非易事, 是否有一种高效快速准确的方法将相连的管线整合到一起呢? 此文简单介绍从设计管道表、管线目录入手使用 EXCEL VBA 进行管道试压包划分的应用思路和使用方法。

[关键词] EXCEL; VBA; 设计管道表; 管线目录

DOI: 10.33142/sca.v2i6.951

中图分类号: TP311.52

文献标识码: A

Using EXCELVBA to Divide Pipeline Pressure Test package

LUAN Deheng

Sinopec 10th Construction Co., Ltd., Qingdao, Shandong, 266000, China

Abstract: The division of pressure test package is a key process in pipeline installation project. The principle of division is according to the medium, pressure, material and so on of the pipeline system. The grade or flaw detection ratio of the pipeline in the same pressure package can be different, but the test pressure should be the same or similar. The conventional pressure package partition is generally divided by referring to the flow chart to extract the connected single-line diagram. If you want to find a line on the process chart board, it is not convenient to find even the electronic version (PDF format file). Therefore, the flow chart division test package is prone to leakage, and the remaining unfinished single line diagram needs to be checked again. If you do not refer to the flow chart directly using the single line diagram for pressure test package division, you need to follow the directory of the single line diagram from the first line to start one by one comparison. Although there is a directory reference, the pipeline division can be done without omission, but the device pipe network is intricate, branch connections jump frequently. It's not easy to fit all the connected lines together. Do you have a high-efficiency, fast and accurate way to fit the connected lines together? This paper briefly introduces the application idea and method of using EXCEL VBA to divide the pipeline pressure test package from the design pipeline table and pipeline catalog.

Keywords: EXCEL; VBA; design pipeline table; pipeline catalog

1 EXCEL VBA 的应用思路

1.1 引言

管线试压包划分前需参照设计管道表对管线目录进行整理, 将每条线对应的相关信息通过 VLOOKUP 公式导入到一个总表里。这样所有的信息都体现在一个 EXCEL 表中, 之后的试压包包号等信息按列添加进总表即可。总表完善后可以按需要导入到焊接纪录总台帐或单线图材料表中, 从试压包层面对焊接台帐和材料进行细化跟踪。

通过管道表可以看出管线的大体走向, 试压相关数据, 管线的起点和终点等数据。只是起点和终点并不明确, 这正是我们下一步需要细化的工作。

管线目录中的管线号是完整管线号, 除了反应管段号信息, 还能体现出主管直径、材质、管道等级、保温保冷要求等。可以使用文本截取公式 (LEFT、MID 等) 或使用 EXCEL 表的分列功能将管段号提取出来, 从而与设计管道表进行关联对应。这样我们需要的总表就基本成型了。

1.2 管道起始点细化工作

试压包的划分实际上只用到三列数据, 管段号、起点管段号、终点管段号。在按试压包抽图时, 当遇到管网复杂、分支较多的管线 (如蒸汽线、氮气线等) 管线号跳跃分布, 找东找西, 没有次序会容易找重找乱。

当按照单线图目录逐条捋线时, 我们可以忽略分支线造成的管网复杂的影响。只需关注该管线的首和尾, 把管线

的起点和终点以管段的形式填入到对应的表格中。分支线可以暂时不看因为分支线必定在管线目录里，后面捋到分支线时，分支线的起点自然是主线，这样两者之间就连在一起了。

同上方法按照管线目录顺序将管线的起点和终点全部填完，遇到没有起点或终点的情况不需做标记，留空白即可。示例（见表 1）。

表 1

管段号	起点	终点
CWR01-111116	CWR01-111120	
CWR01-111118	CWR01-111120	
CWR01-111120	CWR01-111120	
CWR01-111120	CWR01-311151	
CWR01-111120	CWR01-111120	
CWR01-121301	CWR03-121311	
CWR01-121336	CWR01-131047	CWS01-121335
CWR01-121338	CWR01-131047	CWS01-121337
CWR01-121340	CWR01-131047	CWS01-121339
CWR01-121342	CWR01-131047	CWS01-121341
CWR01-121344	CWR01-131047	
CWR01-122301	CWR01-123014	CWR03-122311
CWR01-122326	CWR01-131047	
CWR01-122328	CWR01-131047	CWS01-122327
CWR01-122330	CWR01-131047	CWS01-122329
CWR01-122332	CWR01-131047	CWS01-122331
CWR01-122334	CWR01-131047	
CWR01-123014	CWR01-121301	CWS01-123016
CWR01-123302	CWR01-131047	CWS01-123301

管段号对应的起点终点可以单独放到另一张表格里备用，后续运行 VBA 程序时会用到。

1.3 EXCEL VBA 的应用思路

1.3.1 过程分析

通过观察图 3 表格可以看出，在表格中同一行单元格中的管线必须是相连的，并且是直接相连。不在同一行的管线也可能出现与同一条管线相连的情况，这种情况我们可以称之为间接相连。间接相连中一种复杂的情况就是树状多层次多分支结构。

单凭人工操作想要在 EXCEL 表格里找全一个包的管线无疑是一个繁重的工作。既然把相关的管线号全部输入到了 EXCEL 表里，那么试压包划分的工作到此阶段就变成了纯 EXCEL 表的数据统计与汇总工作了，对于 EXCEL 来说，管线号只是单元格中的一组数据常量而已。既然是常量，又符合一定的规律，就一定会存在一个方法实现一键操作，将符合规律的管线集合起来。

试压包划分就是赋予某个管线一个试压包号，可以想象试压包是一个集合体，只要与之相连的管线。都在虚拟空间集中到一个集合体里，最终形成多少个集合体，就是多少个试压包。从表格第一行开始向下扫描，遇到符合条件的单元格就将其加入到试压包集合体里，直到扫描到最后一行。符合条件的定义：以表格第 2 行为例（排除标题行），在后面某一行的三个单元格其中的任何一个单元格与第 2 行三个单元格中的任意单元格相同（空白单元格除外），那么在这两行中的管线就在一个试压包里，当确认结果证实，这些管线在一个包里后，将它们做为条件继续向下扫描。判别条件从一开始只有第 2 行，逐渐扩大。条件越多，符合要求的就越多，试压包的管线就越多。

然而扫描完毕之后试压包并没有完全分完，前文提到过，直接相连和间接相连。从头到尾扫描过的线只能保证把直接相连的管线找到，间接相连的管线如果是简单的一层顺延分支结构也能找到，但是如果是向上返回跳跃连接的管线是找不到的，因为当时扫过该位置的时候，相应的条件并没有加入到集合里。这时候解决方法其实很简单，就是保

留条件集合不清空再从头扫描一遍。

分析到这里，可以确定要完成这个分包程序需要多层次多条件循环判断。

1.3.2 VBA 介绍

VBA (Visual Basic for Applications) 是可视直译式程序设计语言 (Visual Basic) 的一种宏语言，是微软开发出来在其桌面应用程序中执行通用的自动化任务的编程语言。主要能用来扩展 Windows 的应用程序功能，特别是 Microsoft Office 软件。也可说是一种应用程式视觉化的 Basic 脚本。该语言是微软公司开发的的应用程序共享一种通用的自动化语言。

EXCEL 中看似复杂的概念和代码都可以从简单的宏录制、VBA 编程基础语句等方面入手，VBA 方法适合想提高工作效率的办公人员，特别是需要经常处理、分析大量数据的相关人员。对于简单的一次性操作，如选择性黏贴可以直接使用宏录制的方法将自己的操作录下来，设置好快捷键便可实现一键操作。

2 VBA 代码编写过程

2.1 循环语句

现在开始使用代码实现划分试压包，首先是循环语句，在 VBA 中循环语句基本包含以下几种：

While Wend

While 判断条件

循环执行主体

Wend 语句结束

Do Loop Until

Do

循环执行主体

Loop Until 直到达到某个条件为止

For Next 或者

For counter=start To end 定义一个变量的循环范围如 i=1 to 10

循环执行主体

Next

循环语句结构是程序语言的三种基本结构之一，无论是什么编程语言基本结构只有三种：顺序、选择、循环。

顺序结构：按照语句顺序顺次执行，如赋值、过程调用语句等。

选择结构：又称分支结构，根据条件选择执行路径；如 If Then 或者是 Select Case。

循环结构：重复执行某一段程序语句，如 For Next。

2.2 数组集合

在了解了循环语句之后，我们遇到的下一个难点就是如何做出一个集合来存放符合条件的管线。在 VBA 中有一种数组赋值的方法。数组就是变量的集合，可以是 1 维，可以是 2 维，数组共享一个名字即数组名，定义数组同声明其他变量的方式一样，不同的是数组有大小，有维度，是变量的集合体。

在 VBA 中有一种特殊的数组叫做字典对象 (Dictionary)，VBA 字典法是 VBA 应用的高阶武器，字典对象是 EXCEL 自带的外部引用程序，使用之前需要进行后期绑定。字典可以看成是一个二维数组，而 EXCEL 表中的单元格也可以用二维数组来表示（行和列）。

字典对象有 6 个方法和 4 个属性：

6 个方法：Add/Keys/Items/Exists/Remove/RemoveAll

4 个属性：Count/Key/Item/CompareMode

字典对象的第一列是关键字 Key 列，key 具有唯一不重复性，是字典对象最重要的特性。在划分试压包程序中我们将使用字典方法中的 Keys 关键字/ Exists 存在/Count 计数这三种方法。

2.3 阶段总结

通俗的讲，划分试压包程序就是使用循环语句和 VBA 字典法将相连的在一个包中的管线临时放在一个虚拟空间集合中，给每个集合编上号就是试压包号。编程的过程是一个试验的过程。在不断的调试运行中，添加一些细节，完善相关的语句从而实现相关的功能。

3 试压包划分程序代码详解

试压包划分程序代码主要分为两部分：单包抽取程序和主体循环程序。

3.1 单包抽取代码分段释义

3.1.1 第一段

Sub 单包抽取()

```
Dim dic1 As Object, i, arr, dic2 As Object, dic3 As Object, m, n
```

```
m = 2
```

```
i = 2
```

```
arr = Range("A1:N" & [a65536].End(3).Row)
```

以上代码定义了程序名称和用到的一些变量名，变量名主要用于循环累计，arr 定义为当前表格 A 列到 N 列的所有有效单元格，之后引用某个单元格时可以用 arr（行，列）来表示，如 arr（1,2）表示位于第一行第二列的单元格。

3.1.2 第二段

```
Set dic1 = CreateObject("Scripting.Dictionary")
```

```
Set dic2 = CreateObject("Scripting.Dictionary")
```

```
Set dic3 = CreateObject("Scripting.Dictionary")
```

第二段是设定三个字典数组（dic1, dic2, dic3），用于存放符合条件的单元格，该语句是字典法外部引用固定格式，与声明变量意义相同。其中两个字典用于存放管线号，另一个用于存放试验压力相关的数据，其中的作用后面会提到。

3.1.3 第三段

```
Do
```

```
While arr(m, 4) = ""
```

```
If arr(m, 1) <> "" Then
```

```
dic2(arr(m, 1)) = dic2(arr(m, 1)) + 1
```

```
dic3(arr(m, 1)) = dic3(arr(m, 1)) + 1
```

```
ElseIf arr(m, 2) <> "" Then
```

```
dic2(arr(m, 2)) = dic2(arr(m, 2)) + 1
```

```
dic3(arr(m, 2)) = dic3(arr(m, 2)) + 1
```

```
ElseIf arr(m, 3) <> "" Then
```

```
dic2(arr(m, 3)) = dic2(arr(m, 3)) + 1
```

```
dic3(arr(m, 3)) = dic3(arr(m, 3)) + 1
```

```
End If
```

```
Exit Do
```

```
Wend
```

```
m = m + 1
```

```
Loop
```

第三段使用 Do Loop 循环嵌套 While Wend 循环语句，遍历有效单元格，arr（m, 4）表示从第一行开始扫描第 4 列单元格中的数据。已 m 初始值为 2 为例，第三段的意思是第二行中，如果试压包列是空白，就把管线号、起点、终点三个同一行中，且不是空值的单元格存入字典 2 和字典 3。

3.1.4 第四段

```
Do
```

```
For i = 2 To UBound(arr)
```

```
If dic2.Exists(arr(i, 1)) Then
```

```
If arr(i, 2) <> "" Then
```

```
dic2(arr(i, 2)) = dic2(arr(i, 2)) + 1
```

```
End If
```

```
If arr(i, 3) <> "" Then
```

```
dic2(arr(i, 3)) = dic2(arr(i, 3)) + 1
```

```
End If
```

```
End If
```

```
If dic2.Exists(arr(i, 2)) Then
If arr(i, 1) <> "" Then
dic2(arr(i, 1)) = dic2(arr(i, 1)) + 1
End If
If arr(i, 3) <> "" Then
dic2(arr(i, 3)) = dic2(arr(i, 3)) + 1
End If
End If
If dic2.Exists(arr(i, 3)) Then
If arr(i, 1) <> "" Then
dic2(arr(i, 1)) = dic2(arr(i, 1)) + 1
End If
If arr(i, 2) <> "" Then
dic2(arr(i, 2)) = dic2(arr(i, 2)) + 1
End If
End If
Next
```

第四段开始使用 Do 循环囊括内部的三个 For 循环,上面这部分为第一个 For 循环,三个 For 循环内容结构基本一致。

意思是:根据第三段存入字典的数据,即以第二行中的相关管线号做为条件,依此向下逐行扫描,如果已经存在(Exists),证明该行中的管段号含有第二行中的任意一个管段号,使用 IF 语句判断一旦成立,就同时把该行的三个管段号也加入到字典 2 里。这样从上到下,把所有与第二行直接相关的行中的管线全部存到了字典 2 里。由于字典的关键词 key 具有唯一不重复性,所以不需担心遇到重复值,字典遇到重复值会自动删除。

3.1.5 第五段

```
For i = 2 To UBound(arr)
If dic3.Exists(arr(i, 1)) Then
If arr(i, 2) <> "" Then
dic3(arr(i, 2)) = dic3(arr(i, 2)) + 1
End If
If arr(i, 3) <> "" Then
dic3(arr(i, 3)) = dic3(arr(i, 3)) + 1
End If
End If
If dic3.Exists(arr(i, 2)) Then
If arr(i, 1) <> "" Then
dic3(arr(i, 1)) = dic3(arr(i, 1)) + 1
End If
If arr(i, 3) <> "" Then
dic3(arr(i, 3)) = dic3(arr(i, 3)) + 1
End If
End If
If dic3.Exists(arr(i, 3)) Then
If arr(i, 1) <> "" Then
dic3(arr(i, 1)) = dic3(arr(i, 1)) + 1
End If
If arr(i, 2) <> "" Then
dic3(arr(i, 2)) = dic3(arr(i, 2)) + 1
End If
```

```
End If  
Next
```

第五段内容同第四段，只是把管段号存入另一个字典（字典3）中，这里设定另一个字典作用就是进行比对，前面提过，循环语句作用是遍历所有单元格，遍历的意思是按顺序从上到下扫描。即每个单元格只扫描一次，并不会重复扫描相同的单元格。

3.1.6 第六段

```
For i = 2 To UBound(arr)  
If dic2.Exists(arr(i, 1)) Then  
If arr(i, 2) <> "" Then  
dic2(arr(i, 2)) = dic2(arr(i, 2)) + 1  
End If  
If arr(i, 3) <> "" Then  
dic2(arr(i, 3)) = dic2(arr(i, 3)) + 1  
End If  
End If  
If dic2.Exists(arr(i, 2)) Then  
If arr(i, 1) <> "" Then  
dic2(arr(i, 1)) = dic2(arr(i, 1)) + 1  
End If  
If arr(i, 3) <> "" Then  
dic2(arr(i, 3)) = dic2(arr(i, 3)) + 1  
End If  
End If  
If dic2.Exists(arr(i, 3)) Then  
If arr(i, 1) <> "" Then  
dic2(arr(i, 1)) = dic2(arr(i, 1)) + 1  
End If  
If arr(i, 2) <> "" Then  
dic2(arr(i, 2)) = dic2(arr(i, 2)) + 1  
End If  
End If  
Next
```

第六段内容与第四段内容完全相同，既然 For 循环一次只能找到直接相连的管线，那么再循环一次就必然会找到间接相连的第二层分支线，并且排序在第二层分支线之后与之直接相连的管线也会被找到。

3.1.7 第七段

```
Loop Until dic3.Count = dic2.Count
```

第七段对应第四段开头的 Do 循环，作为整体大循环的收尾。上面三个 For 循环在 Do 循环里不断重复运行，像一个雪球越滚越大。直到把所有在一个网络里的管线全部找齐。那么到何时停止呢，这正是设置另一个字典的功能。两个字典的计数（Count）属性起到了作用，新的数据存入字典的同时也在计数。如果还有新的管线没有加入到字典里，那么经过下一次循环，总数必定还要增大。换言之当两个字典总数相同时证明试压包管线已找齐全。

这也是为什么第六段字典2的 For 循环需再运行一次，这样两个字典总数才会产生差别，才能产生比较。直到两者的总数相等，结束循环。

3.1.8 第八段

```
For i = 2 To UBound(arr)  
If dic3.Exists(arr(i, 1)) Then  
Range("D" & i) = "TP-X"  
dic1(arr(i, 5)) = dic1(arr(i, 5)) + 1
```

```
End If
Next
```

经过前几段核心程序算法的计算，接下来需要得出结果了。第八段的意思是使用 For 循环把第一列（管段号列）所有跟字典中已经存在的数据相同的管段号连系起来，相当于使用了 VLOOKUP 索引功能，选择第一列是因为管段号列是以单线图目录为基准得来的，反映出的是全部的管线。如果索引符合了条件，就在对应行的 D 列（试压包列）填入固定试压包号“TP-X”。同时将对应的第 5 列，即对应的同行的试验压力存入一个新的字典 dic1。新的字典 1 的作用将在下一段代码中体现。

3.1.9 第九段

```
For i = 2 To UBound(arr)
If dic1(arr(i, 5)) <> Application.max(dic1.items) Then
If Range("D" & i) <> "" Then
Range("E" & i).Interior.ColorIndex = 3
End If
End If
Next
```

第九段同样使用 For 循环，对上一段的试验压力进行分析，VBA 字典的一个经典用法就是去除重复项，分类汇总。那么通过上面的程序，对同一个试压包中的管线试验压力进行比对，max(dic1.items) 意为找出重复次数最多的值，对于同一个试压包，试验压力应该完全相等。（Interior.ColorIndex = 3）意为把单元格填充为红色。第九段其实是一个辅助功能，意思是突出显示试验压力的异常情况。便于观察同一个试压包中的压力分布，进而确认试验压力不统一是何原因，是否需要分开试压。如果压力差是在可允许的范围内，便可以并入一个试压包中。

3.1.10 第十段

```
Set dic2 = Nothing
Set dic3 = Nothing
Set dic1 = Nothing
End Sub
```

清空所有字典数组，单包抽取程序运行结束。

3.2 主体程序代码分段释义

单包抽取程序做为试压包划分程序的核心程序运行了一次之后只会找出一个试压包，想要继续再找下一个包，可想而知需要在程序外围再套上一层循环。由于单包抽取程序中条件循环嵌套过多，为了避免逻辑混乱，特单独再做一个主体程序。然后使用 call 命令随时调取单包抽取程序即可。

3.2.1 第一段

```
Sub 主体程序()
Columns("D:D").ClearContents
Columns("E:E").Interior.ColorIndex = 0
[D1] = "试压包号"
```

第一段定义主体程序，将 D 列内容清空，将 E 列颜色清空，D1 单元格赋值内容为“试压包号”。

3.2.2 第二段

```
Dim arr, m, n
m = 2
n = 1
arr = Range("A1:N" & [a65536].End(3).Row)
声明定义一些变量，程序开始部分的固定格式。
```

3.2.3 第三段

```
For m = 2 To UBound(arr)
If arr(m, 4) = "" Then
```

```
Call 单包抽取  
Range("D1:D" & [a65536].End(3).Row).Select  
Selection.Replace "X", n, 2  
[D1].Select  
n = n + 1  
End If  
Next  
End Sub
```

第三段使用 For 循环判断第 4 列试压包列，一开始没有运行单包抽取程序时，全部是空白，那么就默认从第 2 行开始。之后再运行时，因为已经不同程度的填入了一些试压包号，所以遇到填入内容的单元格程序会自动跳过。由于试压包号固定写入的是“TP-X”，调用完单包抽取后，需要在 D 列区域内把“X”替换成数字编号变量 n，只要发生一次替换，n 循环加 1。这样就实现了试压包流水号的填入。全部空白填完后主体程序运行结束。

4 结论

随着办公自动化在企业中的普及，作为 microsoft office 重要组件之一的 EXCEL 越来越得到广泛的使用。日常办公，批处理文档、数据等。EXCEL 是日常工作、学习和生活中信息处理的好帮手。也是现代办公室不可缺少的软件之一。

VBA 语言是机器语言，看似高深，实际也是一种语言，跟我们人与人的沟通交流一样。不要被自己想象的困难吓倒。从基础开始学习使用 VBA，如 msgbox(“你好，世界!”)：输出对话框“你好，世界!”。从一个简单的语句开始，一个简单的语句结束。正是这许许多多的简单语句互相配合使用解决了 EXCEL 数据处理中的一个又一个难题。

解放思想永无止境，无需被固有的传统观念限制了我们的想象。时代在进步，社会在发展，把复杂的问题简单化、步骤化，把简单的问题程序化、逻辑化。只要思想不滑坡，办法总比困难多。任何问题只要认真分析清楚后，总能找到相应的解决办法，把不可能变为可能。

试压包划分程序使用时，需将整理好的数据复制到模板的相应位置，包括管段号、起点、终点、试验压力。之后点击按钮即可运行程序。然后根据需要筛选加工数据即可。

[参考文献]

- [1] 罗刚君. Excel VBA 程序开发自学宝典[M]. 北京: 电子工业出版社, 2014.
- [2] 伍远高. Excel VBA 开发技术大全[M]. 北京: 清华大学出版社, 2014.
- [3] 周庆麟. Excel 应用大全[M]. 北京: 人民邮电出版社出版的图书, 2008.
- [4] Excel 之家. 别怕, Excel VBA 其实很简单[M]. 北京: 人民邮电出版社出版的图书, 2012.
- [5] Excel 之家. Excel VBA 实战技巧精粹[M]. 北京: 人民邮电出版社出版的图书, 2013.

作者简介: 栾德珩(1988-), 本科学历, 目前职称: 工程师。