

# 第四章、LIS 接口编写详解

## 目录

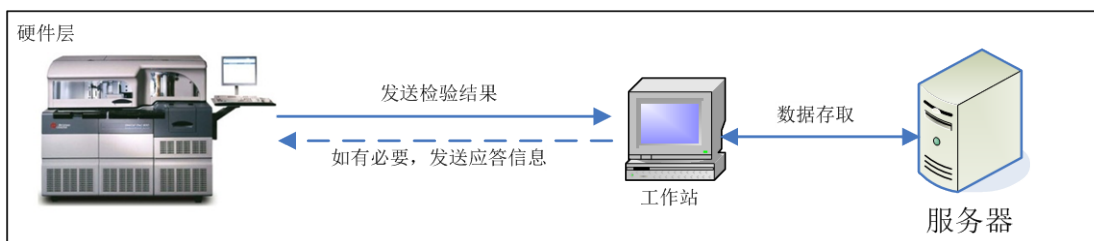
4.1、接口概述与通讯流程.....	2
4.2、接口函数详解.....	3
4.2.1 Analyse过程.....	3
4.2.2 ResultFromFile函数.....	3
4.2.3 CmdAnalyse函数.....	3
4.2.4 GetStartCmd函数.....	4
4.2.5 GetEndCmd函数.....	4
4.2.6 GetAnswerCmd函数.....	4
4.2.7 SendSample函数.....	4
4.3、数据格式详解.....	5
4.3.1 检验结果数据格式.....	5
4.3.2 微生物结果数据格式.....	6
4.3.3 图形结果数据格式.....	6
4.3.3.1 0=直方图 .....	7
4.3.3.2 1=散点图 .....	7
4.3.3.3 2=粘度特征曲线 .....	8
4.3.3.4 3=血沉曲线 .....	8
4.3.3.5 4=PLT双曲线图.....	9
4.3.3.6 5=带界标直方图 .....	10
4.3.3.7 图形文件.....	11
4.3.4 酶标仪的数据格式.....	11
4.3.5 流模式数据格式.....	12
4、接口开发详解.....	12
4.4.1 接口开发步骤.....	12
4.4.2 无图形仪器接口代码分析.....	13
4.4.3 图形代码分析.....	14
4.4.4 酶标仪接口分析.....	17
4.4.5 双向接口分析.....	23
4.4.6 微生物仪器接口分析.....	28
4.5、接口调试详解.....	30

## 4.1、接口概述与通讯流程

LIS 接口是仪器和检验技师工作站沟通的桥梁，它的主要功能是将接收到的原始数据转为我们能够识别的格式，将要向仪器发送的信息转为仪器能识别的指令。

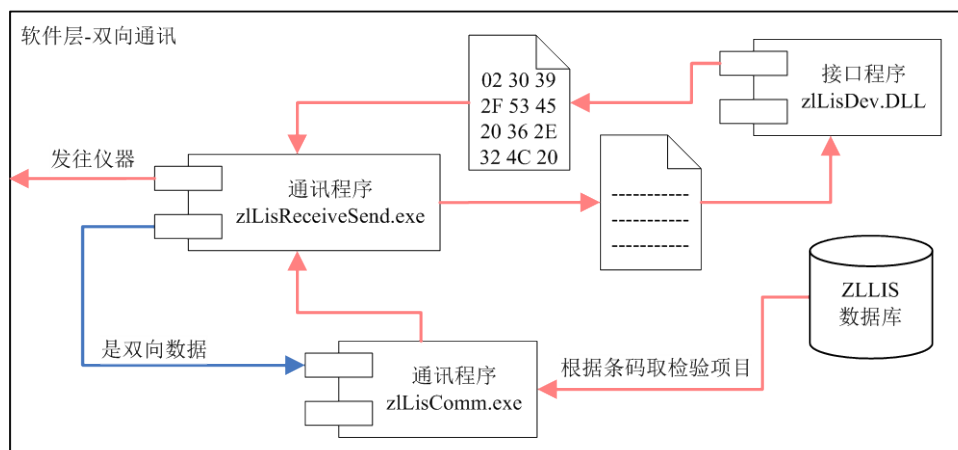
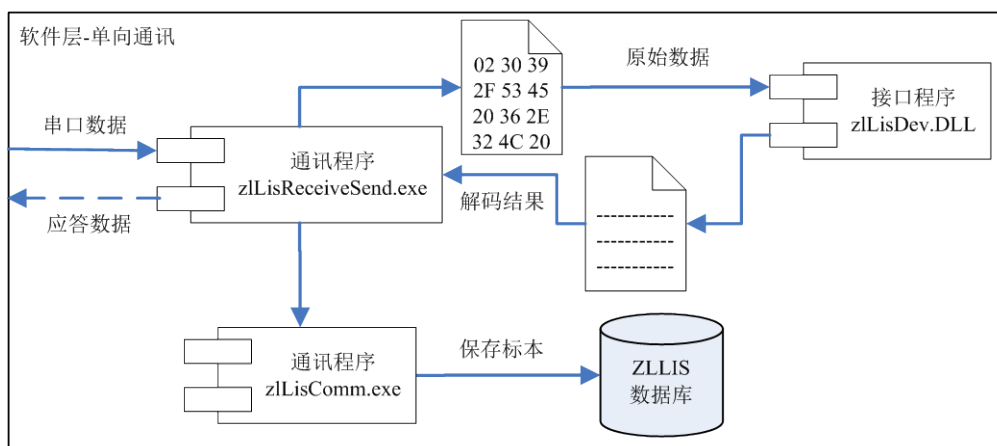
首先，我们介绍一下 LIS 和仪器的通讯流程。LIS 和仪器的通讯流程，可分为单向通讯和双向通讯，判断标准为是否需要发送指令控制仪器。

不论单向，双向通讯，仪器和计算机的连接方式都是相同的，如下：



单向，双向通讯只在软件层面上体现，单向通讯只接收仪器发回的数据，不对仪器发送控制指令；双向通讯即根据仪器发出的询问指令，从 LIS 中取得数据，并把控制指令传给仪器，双向通讯包括单向通讯的流程，是单向通讯流程的进一步扩展。

如图：



## 4.2、接口函数详解

编写 LIS 接口，就是编写约定好的几个函数，下面就详细介绍这几个函数的名称，功能，入参，出参。

### 4.2.1 Analyse 过程

功能

用于串口或网络通讯的接口，当收到信息时调用，用于解析收到的信息，如有解码结果则以字符串方式返回，每组检验结果以||分隔。

入参

strSource: 接收到的仪器原始数据。

出参

strResult: 检验结果数据字符串，每组结果以||分隔

strReserved: 保存本次未解析完的数据。

strCmd: 保存应答指令。当收到双向开始指令时，须以“1|”开头返回，仪器说明书上的指令放在“1|”之后返回，收到的不是双向指令时，按仪器说明书的的应答符返回。另外，如果仪器说明书中要求应答的指令包含“|”，但此指令又不是双向指令，则须以“0|”开头，再加上说明书上的指令返回。

### 4.2.2 ResultFromFile 函数

功能

用于导文件方式的接口，当在技师工作站菜单中选择了“导入”功能时调用，此函数根据传入的参数从文件中获取结果，并以字符串数组方式返回解码结果。

入参

strFile: 文件名

lngSampleNO: 标本号，为-1 表示选取所有时间范围内的标本

dtStart: 开始时间。如果指定了标本号，则只提取该参数当天的这个标本（dtEnd 无效）

dtEnd: 结束时间。只有当选取多个标本时，该参数有效。

返回

检验结果数组

### 4.2.3 CmdAnalyse 函数（酶标仪）

功能

酶标仪接口特有函数，当在酶标仪界面上进行操作时调用，用于转换酶标仪的固定和自定义指令为具体仪器的指令。固定指令与自定义指令，我们在介绍酶标仪接口实例时再详细说明。

入参

strCmdType: 待解析的指令

返回

酶标仪的指令

## 4.2.4 GetStartCmd 函数

功能

在通讯程序第一次启动接口时会调用此函数，用于获取设备开始发送命令。

入参

无

返回

设备开始时需要发送的指令，没有则返回空

## 4.2.5 GetEndCmd 函数

功能

在接口关闭时会调用此函数，用于获取设备结束发送命令。

入参

无

返回

设备结束时需要发送的指令，没有则返回空

## 4.2.6 GetAnswerCmd 函数

功能

当在 zlliscomm 的参数设置中指定了要定时应答参数时调用，用于获取设备定时发送的指令。

入参

无

返回

设备每隔一定时间就需要发送的指令，没有则返回空

## 4.2.7 SendSample 函数（双向）

功能

当在 Analyse 过程中收到仪器双向通讯开始的指令后，通讯程序根据 Analyse 过程中返回的条码或标本号在 LIS 中提取对应的标本信息，发给 SendSample 函数，函数在收到

标本信息时，将标本信息转换为仪器指令返回。

入参

strSampleInfo: 标本信息，标本之间以||分隔，元素之间以|分隔

第 0 个元素：检验时间

第 1 个元素：样本序号

第 2 个元素：检验人

第 3 个元素：标本

第 4 个元素：是否急诊

第 5 个元素：条码

第 6~9 元素：系统保留

从第 10 个元素开始为需要的检验项目

iSendStep: 0 表示未开始双向通讯或双向通讯已完成，系统在收到仪器的应答后，调用 Analy 过程进行后续处理；不等于 0 则表示需要多步才能完成双向通讯，系统在收到仪器的应答后，仍然调用 SendSample 函数进行处理。

blnSuccess: 是否取消发送。

strResponse: 双向通讯期间，仪器的应答信息

返回

仪器能识别的指令

## 4.3、数据格式详解

### 4.3.1 检验结果数据格式

检验结果就是我们要将收到的原始数据转换成的最终格式，每组检验结果的元素之间以|分隔，下面详细说明（[]中的是可选项）：

元素位置	内容	备注
0	检验时间	YYYY-MM-DD 或 YYYY-MM-DD HH:MM:SS
1	样本序号[^是否急诊^条码]	返回有条码时才使用
2	检验人	一般为空格
3	标本类型	一般填血液，尿液等，微生物接口此处填细菌
4	是否质控品	0-不是 1-是
5, 6	检验项目 1 检验结果 1	微生物接口此处填 抗生素 抗生素结果。 如是酶标结果此处填 通道码 定性结果[^OD^CutOff^sco]
5i,5i+1.....	检验项目 i 检验结果 i.....	
最末结果之后	[图形数据]	图形数据格式详细说明在第 3.3 节，微生物仪器接口无图形数据

根据上面的定义描述，下面结合一个实例来理解数据格式，如图：

日期 标本号 检验人

2008-03-26 16:56:00 0074

尿液 0 COL 淡黄色 CLA 透明 GLU 阴性

标本 是否质控 检验项目1 检验结果1 检验项目2 检验结果2 检验项目3 检验结果3

## 4.3.2 微生物结果数据格式

抗生素的结果格式为，“药敏结果^药敏方法^结果值”。

其中，药敏结果取值为“R,S,I,ESBL,BLAC,R\*”，R 表示耐药，S 表示敏感，I 表示中介，ESBL 表示  $\beta$ —内酰胺酶，BLAC 表示未知，R\*表示结果不明确

药敏方法取值为“1，2，3”。1 表示 MIC，2 表示 DISK，3 表示 K-B。

为了便于理解，下面以一个细菌检验结果的例子说明

日期 标本号 检验人

2009-01-10 1001

细菌 是否质控 抗生素1 药敏结果1 药敏方法1 结果1 抗生素2 药敏结果2 药敏方法2 结果2

ENTCFAI 0 SAM R^1^>16 LVX R^1^>4

## 4.3.3 图形结果数据格式

图像数据跟随在普通检验结果数据之后，使用回车换行符来分隔，图像数据是附加在检验结果之后的数据，如果没有图形数据，不影响检验结果的回传。当有多个图像数据时，图像数据之间使用“^”来分隔。

图像数据的总体格式均为：“图像名称;图像类型;图像数据”，绘图程序根据图像类型，按图像数据绘制图像，而图像数据的格式根据图像类型不同，会有所变化。

目前支持的图像类型有(今后可能会增加):

- 0=直方图
- 1=散点图
- 2=血流变粘度特征曲线
- 3=血沉曲线
- 4=PLT 双曲线图
- 5=带界标支持双曲线，XY 座标刻度值的直方图

100-107=BMP 图形文件

110-117=JPG 图形文件

120-127=GIF 图形文件

200-207=压缩为 ZIP 的 BMP 图形文件

210-217=压缩为 ZIP 的 JPG 图形文件

220-227=压缩为 ZIP 的 GIF 图形文件

100-227 中的最后一位数字是图片在控件中的填充形式，用于显示图片时设置合适的对齐方式。其中，1=居中，2=平铺，3=填充，4=拉伸，5=按宽度拉伸，6=按高度拉

伸, 7= 剪裁

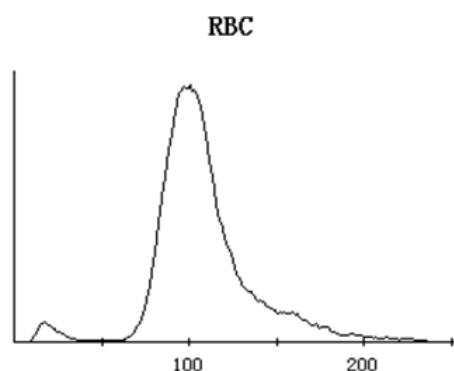
#### 4.3.3.1 0=直方图（没有纵轴刻度）

数据格式：图像名称;0;纵轴高度,横轴长度,刻度 1-显示值,刻度 2-显示值,...;描点数据 1;描点数据 2;描点数据 3...。

例如，有以下数据：

WBC;0;120;262;52-;104-100;156-;208-200;260-;0;0;0;0;0;0;0;1;2;4;5;7;8;10;10;11;10;009;  
009;008;007;007;006;006;006;005;005;004;003;003;002;002;002;002;001;001;001;001;001;  
001;  
001;002;002;003;003;004;005;006;007;009;010;012;016;019;022;028;031;039;044;050;057;066;  
072;080;086;091;099;106;111;115;123;127;132;138;139;140;142;141;141;141;143;139;140;139;1  
36;133;131;126;120;115;108;101;096;092;084;077;072;068;066;063;058;055;053;051;049;045;0  
42;038;036;035;033;031;030;029;028;028;028;026;025;023;022;023;022;022;021;020;020;019;0  
19;018;018;016;016;017;016;017;016;016;016;016;017;016;015;016;014;013;012;012;012;011;0  
10;009;009;009;010;009;007;008;008;008;008;008;007;006;006;006;005;005;004;004;004;0  
04;004;004;005;005;005;004;004;004;004;004;003;003;003;003;003;003;002;003;002;002;0  
02;002;003;003;002;002;002;002;002;002;001;002;002;002;002;002;002;002;002;001;001;0  
01;001;001;001;001;001;000;000;000;000;000;000;000;000;000;000;000;000;000;000;000

绘出的图形如下:



#### 4.3.3.2 1=散点图

散点图以点阵方式保存每一行,使用分号来分隔.有多少个分号就有多少行,每一行有多少个点由每一行的长度来确定,画图的方向是从最上边向下画,如有 65\*65 的图就是从 65 行开始画(最上边开始画)

这个格式的图形，因为缺少界线，不支持彩色等原因，医院用得不多，只有个别接口在用。今后大家在做的时候，尽量不用这个格式，保留它只是为了兼容以前的接口。

这个格式的散点图例图如下:



### 4.3.3.3 2=粘度特征曲线

数据格式：“图像名称;图像类型;座标数据;曲线及描点数据;坐标轴标题数据”。

其中 座标数据的格式为: Y 长度,X 长度|X 座标-X 座标显示的数字,....|Y 座标-Y 座标显示的数字,....

曲线及描点数据的格式为: 粘度曲线 1 的高点和低点座标|粘度曲线 2 的高点和低点座标~低切点坐标,中切点坐标,高切点坐标

坐标轴标题数据的格式为: Y 坐标标题文字,X 坐标,Y 座标~X 坐标标题文字,X 坐标,Y 座标

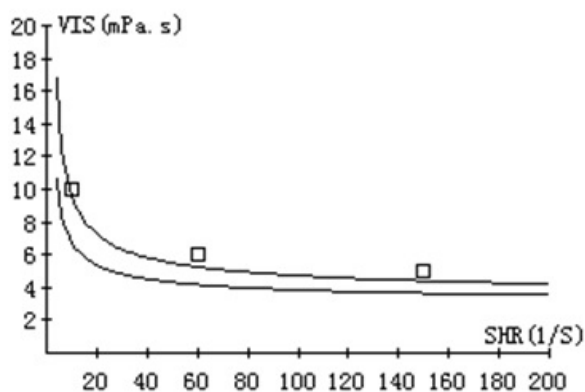
例如, 以下数据:

粘度特征曲

线;2;20,200|20-20,40-40,60-60,80-80,100-100,120-120,140-140,160-160,180-180,200-200|2-2,4-4,6-6,8-8,10-10,12-12,14-14,16-16,18-18,20-20;9.25,10,4.4,150|6.5,10,3.65,150~10-10,60-5.98,150-5.04;VIS(mPa.s),30,20~SHR(1/S),195,1

绘出的图形如下:

粘度特征曲线



### 4.3.3.4 3=血沉曲线

数据格式：“图像名称;3;座标数据;描点数据;坐标轴标题数据”

其中 座标数据: Y 长度,X 长度|X 座标-X 座标显示的数字,....|Y 座标-Y 座标显示的数字,....



绘出的图形如下:



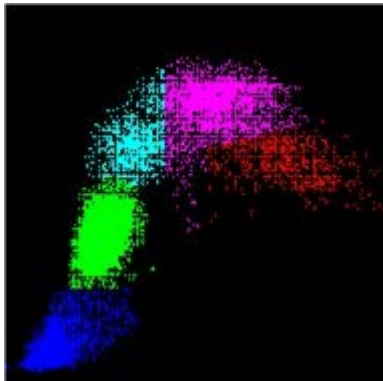
#### 4.3.3.7 图形文件

缺点：（数据量比较大，需要自己写出画图的代码）

图形类型从 100-227 都是指的图形文件，这种格式就是由接口程序完成复杂的图形绘制，如彩色散点图，然后将图形保存为文件，为减小图形文件所占用的空间，后又增加了压缩功能，这一类图形数据的格式均为：“图像名称;图像类型;[图片保存后是否删除];全路径”

例如，有以下数据：RBC;103;1;C:\TEMP\ABC.BMP

对应的图形如下：



#### 4.3.4 酶标仪的数据格式

和其他仪器的结构不同，酶标仪返回的数据格式就是酶标板的数据，共八行，每行 12 个数据，行与行之间用|分隔，每行内的数据用;分隔。如图：

酶标板												
	01	02	03	04	05	06	07	08	09	10	11	12
项目												
A	1	2	3	4	5	6	7	8	9	10	11	12
B												
C												
D												
E												
F												
G												
H												
	01	02	03	04	05	06	07	08	09	10	11	12

例如：

0;0;0;0;0;0;0;0;0;0|0;0;0;0;0;0;0;0;0;0|0;0;0;0;0;0;0;0;0;0|0;0;0;0;0;0;0;0;0;0|0;0;0;0;0;0;0;0;0;0  
;0;0;0;0;0;0;0;0;0;0|0;0;0;0;0;0;0;0;0;0|0;0;0;0;0;0;0;0;0;0|0;0;0;0;0;0;0;0;0;0

### 4.3.5 流模式数据格式

在通信程序中，以单字符方式逐个发送数据，每一个数据范围 0-255（即十六进制的 00-FF）。假设在 WINDOWS 下运行以下程序：

```
Dim i
For i=0 to 255
    MSComm1.Output=chr(i)
Next i
```

希望在接收端得到预期的 0-255 之间的数据，结果却是：前 129 个数据接收正确，为 0-128，后面 127 个数据为 126 个 0 和一个 255，造成这种给果的原因在于中文 Windows 使用的是双字节字符集（DBCS）系统。DBCS 系统使用 0-128 之间的数字表示 ASCII 字符，大于 128 的数字仅作为前导字符，它只是显示是一个非拉丁语系的字符，而并不代表实际意义。上述程序在调用 CHR（）函数时用到了 DBCS 字符集，因此产生了此类错误。那么，如何发送大于 128 的数据呢？答案就是采用流模式来收发数据。

在与仪器通讯的过程中，大部分情况下，我们按字符模式接收和发送数据，但有少部分仪器（主要是含有图片数据的仪器），因为要收发大于 128 的数据，所以必须采用流模式接收和发送数据。流模式收发数据，需用字节数组，为了在现有接口中统一处理，不增加新的函数，所以接收程序内部将字节数组转成字符串，然后把字符串传入接口函数中，接口函数内部仍按字符串方式处理。当然，因为上面提到的原因，为避免数据丢失，传入的字符串没有转换为实际的字符，是一种自定义的格式。这个格式非常简单，它以“,”号分隔每个字符，字符以十六进制的字母表示，例如，收到的数据为“，06”，表示十六进制的 06。下面是这种格式的部分数据示例：

,02,02,02,02,02,05,43,54,52,03,57,42,43,0C,56,61,6C,16,31,36,2E,33,31,08,4C,6F,77,16,34,2E,30,30,08,48,69,67,68,16,31,30,2E,30,30.....

这种格式的数据，将之转为字符，可用 CHR()函数；要将之转为数字，则用 cbyte()函数。大家可参考 KX21，BC5500 等仪器的代码来熟悉这种格式的数据处理，一般来说，公司开发的的这种格式的接口，名称以\_bin 结尾，例如：clsLISDev\_KX21\_bin，clsLISDev\_Mek7222\_Bin 等。

## 4、接口开发详解

### 4.4.1 接口开发步骤

开发接口的步骤一般可以分为以下几步：

1.采集数据（详细步骤见《怎样进行数据采集.doc》）

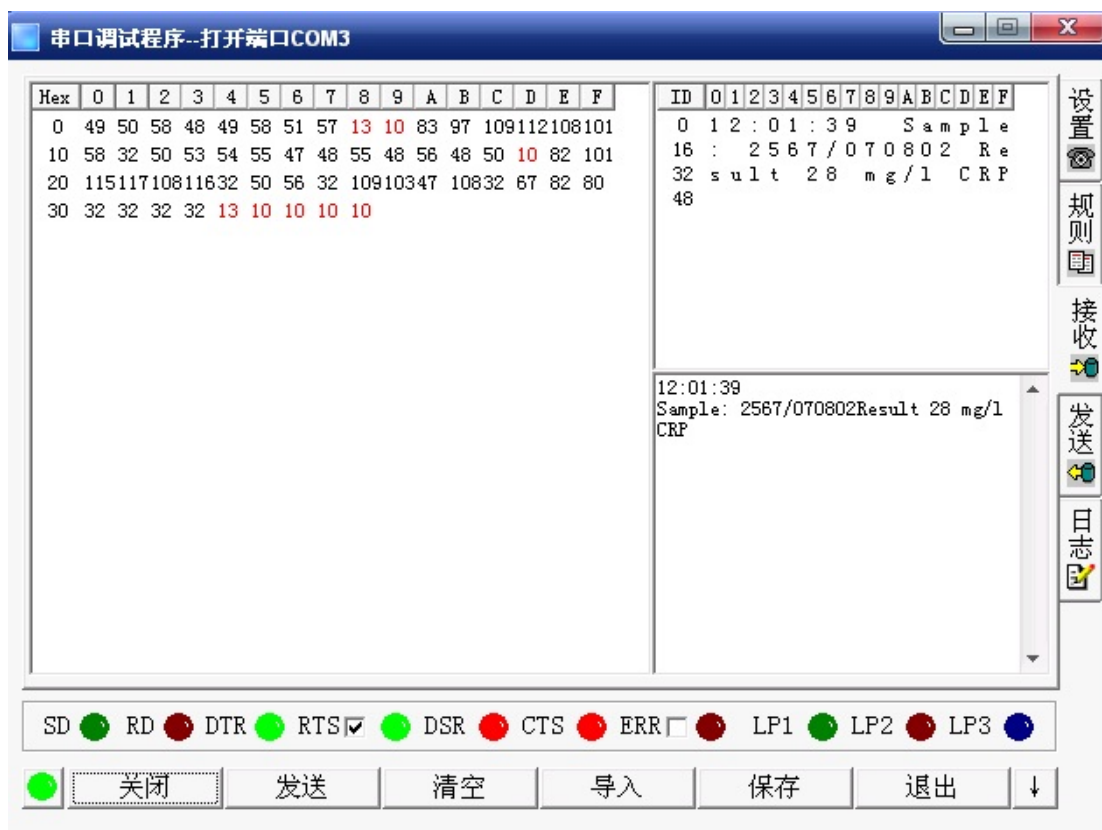
2.分析采集到的数据，找到至少 4 个数据项，“日期，标本号，检验项目，检验结果”。这一步是做仪器接口的难点之一，如果有数据格式说明资料，那么接口开发难度就小，否则就很难甚至做不了。如果仪器数据被厂商加密，又不对我们开放，那么接口就可能没办法做出来，只有另想办法。

3.分析完仪器数据之后，编写接口就容易了，这一步完成从原始数据中提取数据项，按第三章的格式返回即可。

4.调试。（在第五章详细介绍）  
下面我们通过例子来熟悉这几个步骤。

## 4.4.2 无图形仪器接口代码分析

我们采到的仪器数据如下



这个仪器的数据比较简单，我们可以比较直接的从数据中找到“日期，标本号，检验项目，检验结果”，这里就不详细说明了，我们直接进入编码步骤。

这个仪器我们和检验技师交流后得知，只接收 CRP 项目的检验结果，也不需要应答，所以我们只需要编写 Analyse 过程即可，如下：

Option Explicit

Private mstrCode As String '缓存接收到的数据

Public Sub Analyse(ByVal strSource As String, strResult As String, strReserved As String, \_  
strCmd As String)

Dim lng\_S As Long, lng\_E As Long

Dim str\_Sample As String, str\_Day As String

Dim str\_No As String, Str\_Item As String, str\_Value As String

'第一步，缓存接收到的数据

'缓存是因为串口通讯速度较慢，接收数据时，有可能会接收多次才能接收到一个  
'完整的数据

mstrCode = mstrCode & strSource

'第二步，判断是否收到了一个完整的数据，是完整的数据才开始解码，（开始和结束标志）

否则，什么都不做。

lng\_S = InStr(mstrCode, " Sample:") InStr: 查找: Sample:在字符串 mstrCode 中的位置

If lng\_S > 0 Then

    lng\_E = InStr(lng\_S, mstrCode, " CRP")

Else

    lng\_E = InStr(mstrCode, " CRP")

End If

Do While lng\_S > 0 And lng\_E > 0

    '第三步，收到完整的数据，先把这个数据从缓存中取出来。

    str\_Sample = Mid(mstrCode, lng\_S, lng\_E - lng\_S + 3)

    mstrCode = Mid(mstrCode, lng\_E + 3)

    '第四步，取出日期，标本号，项目，检验结果

    str\_Day = Format(CDate(Mid(str\_Sample, 14, 2) & "-" & \_

        Mid(str\_Sample, 16, 2) & "-" & \_

        Mid(str\_Sample, 18, 2)), "yyyy-MM-dd")

    str\_No = Mid(str\_Sample, 8, 5)

    Str\_Item = "CRP"

    str\_Value = Trim(Mid(str\_Sample, 27, 4))

    '第五步 将取出的数据组织成规定格式存入 strResult 变量。

    If str\_Value <> "" And Val(str\_No) <> 0 Then 判断是否有结果和标本号是否为0

        strResult = strResult & "||" & str\_Day & "|" & str\_No & "|血清|0|" & \_

        Str\_Item & "|" & str\_Value     "||" 多个标本

    End If

    '第六步 检查缓冲中的数据，如果还有完整的记录，则继续解码

    lng\_S = InStr(mstrCode, " Sample:")

    If lng\_S > 0 Then

        lng\_E = InStr(lng\_S, mstrCode, " CRP")

    Else

        lng\_E = InStr(mstrCode, " CRP")

    End If

Loop

    '解码完成，将多余的||字符去掉。

    If strResult <> "" Then strResult = Mid(strResult, 3)

End Sub

### 4.4.3 图形代码分析

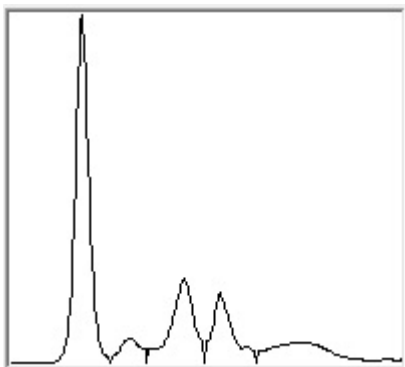
各种仪器的图形数据格式均不相同，图形数据的分析和解码，也是做图形接口的一个难

点，所以，在做仪器接口时，厂商提供的说明书是必不可少的资料。因图形数据格式众多，我们不可能一一介绍，这里就以某仪器的图形数据为例，介绍如何处理仪器的图形数据，并简单介绍下 VB 中如何根据图形数据绘图。

有如下直方图数据：

```
801E001E001D001B001A001A001A001C001D001F001F002100220024002400230020001
C001A001900190019001900190019001800180019001A001C001D001E001E001F001F002
0002100230027002B002E00300032003400350038003C00410046004B00510056005D0063006B
00710077007D00820089008F0096009B00A100A700AD00B100B400B700BA00BC00BD00BD
00BE00BE00C000C100C300C300C400C500C500C300C100BE00BB00B900B700B700B600B5
00B200AE00A900A500A0009C0099009700940090008C00870081007B0078007600760074007
30072807200720075007A0081008B0095009D009E009B00960090008F009700A800BF00DD01
000128014F017A01A401CF01F70225024F0278029502A40294026B022401D501840143011300
F300DE00D200CC80C900CD00D700E700FF011D0143017101A601E40226026D02B002EA03
16032E032D031602EC02B90280024A021601E301B1018001520126010100E100C800B400A50
09A0093008F008D008B008A0089008700840081007F807F007F00820086008D0094009E00AA
00B800C700D600E200EB00EF00EF00E800DE00CE00BC00A80093007D006A0059004D0044
003C0037803400340036003B00450055006A008900B300F1014301B20249030903F7050F0658
07C5094E0AC30C130CF90D430CEB0BF40A8908D00709056603FF02E30213017A010B00C1
008B0064004900360027001C0013000D00080006000500050005000500050005000500040
004000300020001000200030002000100000000000000000000000000000000000000000
00008000
```

我们找用户得到以上数据打印的报告，得知以上数据绘出的图形应该像下面这样：



从仪器说明书上我们得知，图形数据每四位十六进制的数据为一个图像点，从左至右，把得到的十六进制数据转为十进制的数字并连线就可绘出这个图形。

为了方便我们后续使用，我们先编写一个函数，这个函数将图形数据转成字符串，图形数据以；分隔，返回的第一个数据是图片最大高度，绘图时可用于处理画布大小。

转图形数据的函数如下：

```
Private Function SebiaImg(ByVal strimg As String) As String
```

```
    Dim maxVal As Long
```

```
    Do While strimg <> ""
```

```
        '将最大值保存下来，后面绘图时用于设画布大小
```

```
        If maxVal < Val("&H" & Left(strimg, 4)) Then
```

```
            maxVal = Val("&H" & Left(strimg, 4))
```

```
        End If
```

```

        ‘为了图形美观，小于 0 的数据存为 1，大于 4000 的数据舍去
        If Val("&H" & Left(strimg, 4)) < 0 Then
            SebiaImg = "1;" & SebiaImg
        ElseIf Val("&H" & Left(strimg, 4)) < 4000 Then
            SebiaImg = CStr(Val("&H" & Left(strimg, 4))) & ";" & SebiaImg
        End If
        strimg = Mid(strimg, 5)
    Loop
    If Right(SebiaImg, 1) = ";" Then
        SebiaImg = maxVal & ";" & Mid(SebiaImg, 1, Len(SebiaImg) - 1)
    End If
End Function

```

为了验证我们这样处理的图形是否正确，我们可单独编写一个根据以上函数的返回值绘图的过程，这个过程需要传入上面的图形数据，和一个图片控件对象，如下：

```

Private Sub DrawPic(ByVal strData As String, ByRef objPic As PictureBox)
    Dim strImgData As String
    Dim lngMax As Long, varData As Variant
    Dim i As Integer
    ‘转原始图形数据为我们需要的格式
    strImgData = SebiaImg(strData)
    ‘将转换后的数据存入数组，后面循环这个数组描点。
    varData = Split(strImgData, ";")
    ‘设画布的背景色
    objPic.BackColor = vbWhite
    lngMax = Val(varData(0))
    If lngMax < 256 Then lngMax = 256
    ‘设画布的坐标大小
    objPic.Scale (0, 0)-(UBound(varData), lngMax + 50)
    ‘根据图形数据绘图
    For i = LBound(varData) + 1 To UBound(varData)
        If i > LBound(varData) + 1 Then
            ‘后面的点与前一个点连线
            objPic.Line (i - 1, lngMax - varData(i - 1) + 10)- _
                (i, lngMax - varData(i) + 10), vbBlack
        Else
            ‘画第一个点
            objPic.PSet (i, lngMax - varData(i) + 10), vbBlack
        End If
    Next
End Sub

```

如果画出的图形不对，则再分析程序，看是原始数据不是这样处理的还是转数据的程序有问题，总之，要把程序调试为绘出我们需要的图形。



最后，当根据 `SebiaImg` 函数的数据绘出了正确的图形后，我们就可以把 `SebiaImg` 函数拷到我们的接口中，再调整下，按直方图的格式，在此函数的返回串前加上图像名称和坐标数据，就完成了图像数据的解码。

### 4.4.4 酶标仪接口分析

酶标仪接口的开发，主要难点在于对酶标仪工作流程的理解，如果理解了酶标仪的工作流程，有仪器厂商提供的仪器指令资料，那么编写接口就是很轻松的一件事情，下面我们先来了解一下酶标仪的基础知识。

酶标仪主要应用于酶联免疫吸附检测反应中检测吸光度值，测定原理是：在特定波长下，检测被测物的吸光值（OD 值）。酶标仪是检验科常用的检测仪器，一般可通过串口接收酶标仪 RS232 接口输出的数据，将其解析成实际的 OD 值，并利用预设公式对 OD 值进行计算，得到定性结果，然后将结果存到相应的标本中，从而实现酶标仪数据的自动化处理。

具体的工作流程是：

1. 通过 LIS 向酶标仪发送联机指令
2. 发送酶标仪的设置参数，如测量波长，振板频率等。
3. 发送酶标仪进板以及测量指令。
4. 酶标仪读板完成后，数据以 ASCII 格式经 RS232 接口传出。
5. LIS 酶标模块接收原始数据并将其解析成实际的 OD 值
6. 利用对应试剂的 CutOff 公式(临界值计算公式)对 OD 值进行计算，得到对应的定性结果
7. 然后用编号将项目代号、定性结果与相应的标本对应，存到数据库中。
8. 完成酶标仪数据的接收、计算、保存后，向酶标仪发送脱机指令。
9. 在检验技师工作站可通过查询、打印报告的形式输出规范的检验结果。

从以上工作流程，可知 LIS 酶标模块主要完成以下功能：

- 1、ZLLIS 检验技师工作站模块完成通用的酶标仪数据接收、计算功能，并与 ZLLIS 现有的标本管理、查询、报告打印等程序模块集成。
- 2、扩展了现有 ZLLIS 的现有功能，提供了酶标仪专用操作界面，使原来手工方式下繁琐的操作变得简单方便、实现数据的快速计算、酶标仪的数据管理规范。
- 3、增加了专门的酶标仪数据处理模块，将仪器的具体指令放到独立的接口中，达到可适于各种采用 RS232 接口的酶标仪；增加检验项目的酶标公式设置，通过修改公式可得到不同试剂的定性、临界值等检验结果。

酶标仪的操作指令，因每种仪器不可能相同，所以需要将仪器的指令按仪器进行对应，每种仪器一个接口程序，我们放到独立的部件中（ZLLISDEV），将具体仪器的指令处理独立于主程序之外，就达到模块的通用性，今后扩展也很方便。通过分析收集到的资料，酶标仪的指令，我们现阶段将之分为两类，一类为固定指令（操作时只有一个对应的指令，这类指令为系统固定，接口开发者须对这些指令进行转换），一类为自定义指令（操作时有多个可选的指令，这个指令由接口开发人员定义，并在接口中时行转换），这些指令均是在酶标操作界面上执行相应操作时向仪器发出，如下：

固定指令	用途
Begin	开始时，需要向仪器发送的指令
End	结束时，需要向仪器发送的指令

Revert	向仪器发送的应答指令
Play	开始测量时，需要向仪器发送的指令
ReadData	读数据时，需要先仪器发送的指令
Out	控制仪器执行出板操作的指令
In	控件仪器执行进板操作的指令

自定义指令是每个仪器的接口开发者自己定义的指令，例如 MK3 的自定义指令：

自定义指令示例	用途
450;600;900	在仪器上设置波长的指令
1X;2X;3X	设置仪器振板频率的指令
30;50	设置仪器振板时间的指令
连续;步进	设置仪器进板方式的指令
空气;试剂	设置仪器空白形式的指令

还有一种“TimeOut”超时指令，是一种特殊指令，当有些仪器指令执行的时间超过 10 秒时，需要在接口中增加此指令，以避免酶标仪界面出现超时错误，这个指令后面会详细说明。

自定义指令的定义界面在检验仪器管理中如下图：

仪器基本信息

编码 (U) 144

发送时指定杯号 ☐

通讯口 (L) COM1

名称 (N) 科华酶标360

简码 (S) KHMB360

波特率 (2) 4800

类型 (K) 04-免疫学

类别 (G) 2-酶标仪

数据位 (3) 8

连接计算机 (I)

使用 (U) 203-检验科

停止位 (4) 1

通讯程序名 (P) rLLISDev.clsLISDev\_ST360

校验位 (5) None

酶标仪设置

波长 (B) 450

振板频率 (L) 低

振板时间 (T) 30

进板方式 (P) 连续

空白形式 (A) 空气

质控要求：至少每 1 天 进行一次质控，每批检测 1 个水平

仪器QC码

默认试剂来源

默认校准物来源

☐ 对数质控图

自定义指令，如果有多个选项，中间用;分隔

如波长可以设成: 450;650;900

仪器项目通道

序号	编码	中文名	英文名	通道码	精度	加算值	换算比	糖耐项目
1	5061	★乙肝表面抗原	HBsAg	hbsag				<input type="checkbox"/>
2	5062	★乙肝表面抗体	HBsAb	hbsab				<input type="checkbox"/>

3、正式使用时，在酶标仪操作界面中输入标本编号、临界值计算公式（即 CutOff 公式）、阳性对照孔，阴性对照孔，点击“测量”按钮，就可完成酶标仪数据的读取计算（如图 2）。

图 2 操作界面

酶标仪

H) 检验

取消 新增 修改 删除 计算 连接 测量 断开 帮助 退出

显示方式

- ☐ 样本编号
- ☒ 原始OD值
- ☐ OD值
- ☐ 定性值

测量参数

测试时间: 2009-05-27 11:22:23 测试板号: 20090527\_1

波长: 450 参考波长: 450 试剂批号:

振板频率: 低 振板时间: 30 试剂效期:

进板方式: 连续 空白形式: 空气 试剂厂商:

测试项目: ★乙肝表面抗原 (HBsAg) ☒ 单板单项 ☐ 单板多项 测试方法:

阳性公式: OD>NC\*2.1 弱阳性公式: OD>NC\*2.1 CutOff公式: NC\*2.1

模板

选择模板: 保存模板 删除模板 ☒ 横向 ☐ 纵向 开始标本号: OK

微孔板

☒ 普通 (S) ☐ 空白 (BC) ☐ 阴性 (NC) ☐ 阳性 (PC) ☐ 质控 (QC) ☐ 阴性对照小于: 时按设定值计算 重置

	1	2	3	4	5	6	7	8	9	10	11	12
A	0.000	0.005	0.008	0.007	0.007	0.009	3.104	3.218	3.004	3.100	0.013	0.006
B												

连接, 测量, 断开这三个按钮按以下顺序向仪器发送指令(其中 ReadData 是调用接口中的 Analyse 函数来解析数据, 其他的操作调用 CmdAnalyse 函数)。

① 1.连接按钮:

Begin 连接仪器  
Out 弹出微孔板 (出板)

② 2.测量按钮:

in 关闭微孔板(进板)  
450 设置波长(该指令可以自定义,在仪器管理中设)  
低 设置振板频率(该指令可以自定义,在仪器管理中设)  
30 设置振板时间(该指令可以自定义,在仪器管理中设)  
连续 设置进板方式(该指令可以自定义,在仪器管理中设)  
空气 设置空白形式(该指令可以自定义,在仪器管理中设)  
Play 开始测量 (主波长)  
ReadData 读取主波长数据 (开始解析 analyse 函数)  
以下步骤是可选项, 如果没有设置参考波长, 则不执行。  
450 设置参考波长(该指令可以自定义,在仪器管理中设)  
ReadData 读取参考波长数据  
Out 弹出微孔板 (出板)

③ 3.关闭按钮:

End 释放仪器控制

点保存后, 测量的结果就会保存到检验技师工作站中, 用于出报告等, 操作跟其他仪器的检验标本操作一致。

酶标仪的数据很简单, 如下图:



X3	设定振荡的速度为最高值的 33%
Znn	以指定时间振荡微孔板，nn 为振动时间(以秒为单位，00-60)
T	返回仪器的日期时间
N	返回仪器的系列号
V	返回仪器的软件版本号
Q	计算机停止控制，回到键盘操作

说明：

每个命令都必须以<LF>(ASCII=0C)结束，仪器忽略命令前的空格。

对于 P 和 PM 命令，返回的是板格式数据，为如下格式：

```
Data Data Data Data Data Data Data Data Data Data Data Data Data <CR><LF>
Data Data Data Data Data Data Data Data Data Data Data Data Data <CR><LF>
Data Data Data Data Data Data Data Data Data Data Data Data Data <CR><LF>
Data Data Data Data Data Data Data Data Data Data Data Data Data <CR><LF>
Data Data Data Data Data Data Data Data Data Data Data Data Data <CR><LF>
Data Data Data Data Data Data Data Data Data Data Data Data Data <CR><LF>
Data Data Data Data Data Data Data Data Data Data Data Data Data <CR><LF>
Data Data Data Data Data Data Data Data Data Data Data Data Data <CR><LF>
```

在这里，<CR>(ASCII=0D)，Data 共有 6 个字符，第一个字符可是空格或负号，后面五个是数字或小数点，其数据为每一孔的吸光度值。

有了上面的仪器资料，我们编写 CmdAnalyse 函数就很简单了，如下：

```
Public Function CmdAnalyse(strCmdType As String) As String
```

```
    Select Case strCmdType
```

```
        Case "Begin"                                '开始
```

```
            CmdAnalyse = "R" & vbCrLf
```

```
        Case "End"                                    '结束
```

```
            CmdAnalyse = "Q" & vbCrLf
```

```
        Case "Revert"                                '回复命令
```

```
            CmdAnalyse = "OK" & vbCrLf    '仪器应答符
```

```
        Case "ReadData"                              '整板测量
```

```
            CmdAnalyse = "P" & vbCrLf
```

```
        Case "play"
```

```
            Is mintstp=0 then
```

```
                CmdAnalyse = Z(02)|z_end
```

```
        '上面都是固定指令的处理
```

```
        Case "450"
```

```
            CmdAnalyse = ""
```

```
            CmdAnalyse = "F2" & vbCrLf    '450 波片
```

```
        Case "连续"                                    '进板方式连续
```

```
            CmdAnalyse = "E0" & vbCrLf
```

```
        Case "1X"                                        '振板频率 1X
```

```
            CmdAnalyse = "X1" & vbCrLf
```

```
        Case "in_TimeOut"    进版超时 30s, 用原指令+ "_" + 特殊指令 (TimeOut)
```

**CmdAnalyse = 30**

‘上面是部分自定义指令的处理，

‘其他指令大家可以根据上面的资料来自定，并在接口中处理

End Select

End Function

酶标仪在测量数据时，要花一些时间，酶标仪界面上就会显示一个进度。有些指令很快就能回传结果，有些指令则要慢一些，当然也不能排除因外部原因导致收不到结果的情况，所以程序不能一直处于等待状态，为处理这种情况，酶标界面程序内部有超时的处理，发出一个指令后，超过一段时间仍然没有收到检验结果，那么就报超时错误。超时的长短，要根据仪器的指令来设置，所以，接口程序中加入了指令等待时间的处理，就是在原指令的基础上加上 “\_TimeOut”，形成一个新的指令，这个指令的返回值就是超时时间，如果没有得到返回，则用默认的超时时间，也就是说，“\_TimeOut” 指令不是必须的，只有当仪器的指令等待时间较长，超过了默认等待时间，才需要在接口中进行处理。

我们这里仍以 MK3 酶标仪的振板指令为例，说明一下 “\_TimeOut” 指令的处理，代码如下：

**Case "Z30\_TimeOut"**

**CmdAnalyse = “35” & vbCrLf**

上面就是 CmdAnalyse 函数的内容，下面介绍下 Analyse 过程的内容。

Analyse 过程对酶标仪接口来说，就是处理仪器返回的数据（原始数据的格式见前面的说明），将原始数据转为酶标仪的数据格式返回，如下：

**Public Sub Analyse(ByVal strSource As String, strResult As String, strReserved As String, \_  
strCmd As String)**

‘没有这么长时退出

**If UBound(Split(strSource, vbCrLf)) < 8 Then Exit Sub**

‘开始解析

**strResult = strSource**

**For intLoop = 5 To 2 Step -1**

**strResult = Replace(strResult, Space(intLoop), Space(1))**

**Next**

**strResult = Replace(strResult, "OK" & vbCrLf, "")**

**strResult = Replace(strResult, " ", ";")**

**strResult = Replace(strResult, "-", ";-")**

**strResult = Mid(strResult, 2)**

**strResult = Replace(strResult, vbCrLf, "|")**

**strResult = Replace(strResult, "|;", "|")**

**End Sub**

至此，酶标仪的接口就开发完成了。



#### 4.4.5 双向接口分析

双向接口的开发，需要编写一个函数和一个过程，Analyse 过程和 SendSample 函数。前面我们介绍过，双向接口是单向接口的扩展，所以，我们可以先按单向接口的编写方法，完成 Analyse 过程的编写，再在具有单向接口功能的 Analyse 过程中增加双向部分的代码，就完成了 Analyse 过程的编写。完成了 Analyse 过程的编写之后，我们再编写 SendSample 函数，将技师工作站中传过来的标本信息，转换为仪器能识别的指令返回。这两个函数编写完成之后，就可以上机或者在模拟环境下调试双向接口了。

例如，有某双向仪器，用 HHD 采集的数据片断如下：

Answer: 2009-11-23 9:58:29.74864 (+466.0625 seconds)

05

Request: 2009-11-23 9:58:29.74864 (+0.0000 seconds)

06

Answer: 2009-11-23 9:58:29.78064 (+0.0313 seconds)

```
02 31 48 7C 5C 5E 26 7C 7C 7C 7C 7C 7C 7C 7C 7C .1H|\^&|||
7C 50 7C 7C 0D 03 30 35 0D 0A |P| |..05..
```

Request: 2009-11-23 9:58:29.81164 (+0.0000 seconds)

06

Answer: 2009-11-23 9:58:29.82764 (+0.0156 seconds)

```
02 32 51 7C 31 7C 5E 35 32 37 30 39 31 31 32 33      .2Q|1|^527091123
30 30 37 5E 33 32 36 35 5E 30 5E 36 5E 5E 53 41      007^3265^0^6^^SA
4D 50 4C 45 5E 4E 4F 52 4D 41 4C 7C 7C 41 4C 4C      MPLE^NORMAL||ALL
7C 7C 7C 7C 7C 7C 7C 7C 4F 0D 03 37 34 0D 0A      |||||0..74..
```

Request: 2009-11-23 9:58:29.92064 (+0.0313 seconds)

06

Answer: 2009-11-23 9:58:29.93664 (+0.0156 seconds)

```
02 33 4C 7C 31 0D 03 33  43 0D 0A                      .3L|1..3 C..
```

Request: 2009-11-23 9:58:29.93664 (+0.0000 seconds)

06

Answer: 2009-11-23 9:58:29.93664 (+0.0000 seconds)

04

上面监控到的数据中，兰色部分是仪器发出的数据，红色部色是接收程序返回的应答。通过查通讯手册，我们理解了以上数据的含义是：

仪器到接收程序 <=: 请求通讯 (05)

接收程序到仪器 =>: 收到 (06)

仪器到接收程序 <=: 第 1 段数据 (1H | \....)

接收程序到仪器 =>: 收到 (06)

仪器到接收程序 <=: 第 2 段数据 (2Q|1....)

接收程序到仪器 =>: 收到 (06)

仪器到接收程序 <=: 第 3 段数据 (3L|1....)

接收程序到仪器 =>: 收到 (06)

仪器到接收程序 <=: 发完了 (04)

通过上面的数据和我们查通讯手册得到的信息，我们了解了这个仪器通讯过程，一次完整的通讯是从 05 开始，04 结束，中间的数据是待解析信息，到底 05 和 04 中间的数据是什么，我们这里可以先不管它，后面去查通讯手册。这里，我们先介绍在接口中完成应答部份的功能。如下：

Option Explicit

Private mstrCode As String '缓存接收到的数据

Public Sub Analyse(ByVal strSource As String, strResult As String, strReserved As String, \_  
strCmd As String)

Dim lng\_S As Long, lng\_E As Long

Dim str\_Sample As String, str\_Day As String

Dim str\_No As String, Str\_Item As String, str\_Value As String

'第一步，缓存接收到的数据

'缓存是因为串口通讯速度较慢，接收数据时，有可能会接收多次才能接收到一个完整的数据

mstrCode = mstrCode & strSource

'此处处理应答

If Instr(strSource, Chr(&H05))>0 then strCmd= Chr(&H06)

If Instr(strSource, Chr(&H0A))>0 then strCmd= Chr(&H06)

.....

应答部分完成了之后，我们再来处理收到的数据，通过查手册，我们得知，第 1 段数据和第 3 段数据均是为了保证通讯数据的格式完整而加的数据，其中的信息对我们来说，没有什么用处，对我们有用的是第 2 段数据，即“2Q|1|^527091123007^3265^.....”这一段，这段数据中包含条码，和标本在仪器上的位置信息，我们把条码取出返回给通讯程序，要求通讯程序提供这个条码做什么具体项目的一个信息，把位置信息取出保存，在写 SendSample 函数时要用。例子中的条码是“527091123007”，Analyse 过程中取条码的代码如下：

lng\_S = InStr(mstrCode, Chr(&H5))

lng\_E = InStr(mstrCode, Chr(&H4))

If lng\_S > 0 And lng\_E > 0 Then

str\_Sample = Mid(mstrCode, lng\_S, lng\_E - lng\_S + 1)'截取一段数据

mstrCode = Mid(mstrCode, lng\_E + 1)

'双向通讯- 开始

If str\_Sample Like "\*" & Chr(&H2) & "?Q|\*" Then

var\_line = Split(str\_Sample, Chr(&HA))

For i\_Count = LBound(var\_line) To UBound(var\_line)

str\_Line = var\_line(i\_Count)

If str\_Line Like Chr(&H2) & \_

"?Q|\*SAMPLE^NORMAL||ALL|||||\*" Then

str\_条码 = Split(str\_Line, "|")(2)

mstrRack = Split(str\_条码, "^")(2) & "^" & \_

Split(str\_条码, "^")(3) & "^" & Split(str\_条码, "^")(4)

str\_条码 = Split(str\_条码, "^")(1)

If Val(str\_条码) <> 0 and len(Val(str\_条码))=12 Then



```

        '返回双向通讯要用的条码
        strResult = strResult & "|" & _
        Format(Now, "yyyy-MM-dd") & "|" & Trim(str_条码)
    End If
End If
Next
If strResult <> "" Then
    strResult = Mid(strResult, 3)
    '通知通讯程序，开始双向
    strCmd = "1|" & strCmd
End If
End If

```

通过以下代码，我们取出条码，并通知了通讯程序，开始进入双向流程，下面我们详细介绍双向流程中要调用 SendSample 函数，在写这个函数前，我们先来看下我们采集的数据，这段数据就是我们写这个函数最后要达到的效果。

```

Request: 2009-11-23 9:58:29.21764 (+0.2500 seconds)
05
Answer: 2009-11-23 9:58:29.24864 (+0.0313 seconds)
06
Request: 2009-11-23 9:58:29.26464 (+0.0156 seconds)
02 31 48 7C 5C 5E 26 7C 7C 7C 41 53 54 4D 2D 48 .1H|\^&||ASTM-H
6F 73 74 0D 03 35 39 0D 0A ost..59..
Answer: 2009-11-23 9:58:29.29564 (+0.0313 seconds)
06
Request: 2009-11-23 9:58:29.31164 (+0.0156 seconds)
02 32 50 7C 31 7C 7C 35 32 37 30 39 31 31 32 33 .2P|1||527091123
30 30 37 0D 03 39 43 0D 0A 007..9C..
Answer: 2009-11-23 9:58:29.32764 (+0.0156 seconds)
06
Request: 2009-11-23 9:58:29.42064 (+0.0938 seconds)
02 33 4F 7C 31 7C 35 32 37 30 39 31 31 32 33 30 .30|1|5270911230
30 37 7C 33 32 36 35 5E 30 5E 36 7C 5E 5E 5E 33 07|3265^0^6|^^^3
32 31 5E 30 5C 5E 5E 5E 33 39 31 5E 30 7C 52 7C 21^0\^^^391^0|R|
7C 7C 7C 7C 7C 4E 7C 7C 7C 7C 7C 7C 7C 7C 7C ||||N|||||||
7C 7C 7C 7C 4F 0D 03 30 34 0D 0A ||||0..04..
Answer: 2009-11-23 9:58:29.46764 (+0.0469 seconds)
06
Request: 2009-11-23 9:58:29.46764 (+0.0000 seconds)
02 34 4C 7C 31 7C 0D 03 42 39 0D 0A .4L|1|..B9..
Answer: 2009-11-23 9:58:29.49864 (+0.0313 seconds)
06
Request: 2009-11-23 9:58:29.49864 (+0.0000 seconds)
04

```

有了前面的经验，我们再来看这段数据，就容易一些了，这段数据同样是 05 开始，04

结束，一共向仪器发了 4 段数据，第 1，2，4 段是通讯必须的，只表示信息头和信息尾，这些数据不是我们要分析的重点，我们要分析的重点是第 3 段，经过查通讯手册，我们理解了第 3 段数据的格式：

“3O|1|527091123007|3265^0^6|”这部分中的条码和标本在仪器上的位置信息是仪器发给我们的，在 Analyse 函数中我们保存了下来，现在再把它以这种格式返回给仪器，一般是仪器要根据这些信息来验证是不是它刚才发的信息的应答，“^^^321^0\^^^391^0”是从标本信息中取出来的，表示仪器要做的 2 个项目，项目通道码是 321，391，在知晓了要回传给仪器的格式之后，我们再来看 SendSample 函数的代码，就能理解了，SendSample 函数要处理的数据是“2009-11-23|342|段建忠|血清|0|527091123007||||321|391”，代码如下：

```
Public Function SendSample(strSampleInfo As String, _
                           iSendStep As Integer, _
                           blnSuccess As Boolean, _
                           Optional ByVal strResponse As String = "", _
                           Optional ByVal blnUndo As Boolean = False, _
                           Optional ByVal iType As Integer = 0) As String
    Dim var_标本 As Variant, i_Count As Integer, i_Loop As Integer
    Dim str_记录 As String, str_标本号 As String
    Dim i_急诊 As Integer, str_条码 As String, str_检验项目 As String
    Dim str_标本类型 As String
    Dim str_SendItem As String, str_Line As String

    var_标本 = Split(strSampleInfo, "|")
    SendSample = ""

    blnSuccess = True '设为不取消发送
    If iSendStep = 0 Then
        For i_Count = LBound(var_标本) To UBound(var_标本)
            str_记录 = var_标本(i_Count)
            If Val(Split(str_记录, "|")(1)) = 0 Then
                str_标本号 = Trim(Split(str_记录, "|")(5))
            Else
                str_标本号 = Val(Split(str_记录, "|")(1))
            End If
            str_标本类型 = Split(str_记录, "|")(3)
            i_急诊 = Val(Split(str_记录, "|")(4))
            str_条码 = Trim(Split(str_记录, "|")(5))
            str_检验项目 = ""
            For i_Loop = 10 To UBound(Split(str_记录, "|"))
                If Val(Split(str_记录, "|")(i_Loop)) > 0 Then
                    str_检验项目 = str_检验项目 & "^^^" & _
                        Split(str_记录, "|")(i_Loop) & "^0\"
                End If
            Next
        Next
    End If
```

```

If str_检验项目 <> "" And str_条码 <> "" Then
    str_检验项目 = Mid(str_检验项目, 1, Len(str_检验项目) - 1)
    '第一段
    str_Line = "1H\^&||ASTM-Host" & Chr(&HD) & Chr(&H3)
    str_SendItem = str_SendItem & Chr(&H2) & str_Line & _
        GetCheckCode(str_Line) & Chr(&HD) & Chr(&HA) & ","
    '第二段
    str_Line = "2P|1|" & str_条码 & Chr(&HD) & Chr(&H3)
    str_SendItem = str_SendItem & Chr(&H2) & str_Line & _
        GetCheckCode(str_Line) & Chr(&HD) & Chr(&HA) & ","
    '第三段
    str_Line = "3O|1|" & str_条码 & "|" & mstrRack & "|" & str_检验项目 & _
        "|" & If(i_急诊 = 1, "S", "R")
    str_Line = str_Line & "||||N|||||||O" & Chr(&HD) & Chr(&H3)
    str_SendItem = str_SendItem & Chr(&H2) & str_Line & _
        GetCheckCode(str_Line) & Chr(&HD) & Chr(&HA) & ","
    '第四段
    str_Line = "4L|1|" & Chr(&HD) & Chr(&H3)
    str_SendItem = str_SendItem & Chr(&H2) & str_Line & _
        GetCheckCode(str_Line) & Chr(&HD) & Chr(&HA) & ","

    If SendSample = "" Then
        '加上 消息标志(05, 04)
        SendSample = SendSample & Chr(&H5)
        iSendStep = 1 '开始通信
        mstrWaitSend = mstrWaitSend & "," & str_SendItem & Chr(&H4)
    Else
        mstrWaitSend = mstrWaitSend & "," & Chr(&H5) & "," & _
            str_SendItem & Chr(&H4)
    End If
End If
Next
Else
    If InStr(strResponse, Chr(&H6)) > 0 Then
        '收到 06, 将下一条要发送到仪器的指令取出, 放到 SendSample
        If mstrWaitSend <> "" Then
            If Left(mstrWaitSend, 1) = "," Then
                mstrWaitSend = Mid(mstrWaitSend, 2)
            End If
            If InStr(mstrWaitSend, ",") > 0 Then
                SendSample = Mid(mstrWaitSend, 1, InStr(mstrWaitSend, ",") - 1)
                mstrWaitSend = Mid(mstrWaitSend, InStr(mstrWaitSend, ","))
            Else
                SendSample = mstrWaitSend
            End If
        End If
    End If
End If

```

```

        mstrWaitSend = ""
    End If
End If
End If
‘没有待发送指令，退出双向通讯。不能忘了
If mstrWaitSend = "" Then iSendStep = 0
Exit Function
End If
End Function

```

函数中用到的 GetCheckCode 函数是求验证码的函数，是根据仪器说明书中的计算方法编写而成，如下：

```

Private Function GetCheckCode(ByVal str_in As String) As String
    ‘获取校验码
    Dim lngAsc As Long, i As Integer
    For i = 1 To Len(str_in)
        lngAsc = lngAsc + Asc(Mid(str_in, i, 1))
    Next
    GetCheckCode = Right("0" & CStr(Hex(lngAsc Mod 256)), 2)
End Function

```

## 4.4.6 微生物仪器接口分析

下面我们介绍一下微生物的接口，这个接口是从数据库中读取结果，所以，通过这个例子，大家也可以了解如何写 ResultFromFile 函数。

我们先分析下数据结构，找到我们需要的细菌和药敏结果数据，根据用户提供的微生物报告，我们在 ACCESS 文件中的 ReqToExe，ResultBill，MicResult 表中找到了我们需要的数据，下面我们就可以在 ResultFromFile 函数中写 SQL 从数据库中提取我们需要的信息，然后返回，下面是主要代码：

```

Public Function ResultFromFile(ByVal strFile As String, ByVal lngSampleNo As Long, _
    ByVal dtStart As Date, Optional ByVal dtEnd As Date = CDate("3000-12-31")) _
    As String()
    Dim aResults() As String ‘字符串数组，用于返回检验结果
    Dim cnAccess As New ADODB.Connection
    Dim strFilePath As String, objFile As New Scripting.FileSystemObject
    Dim strSQL As String, rsTmp As ADODB.Recordset, rsAnti As ADODB.Recordset
    Dim strBh As String, strAnti As String, str 结果 As String, str 细菌 As String
    Dim lngResultCount As Long

    strFilePath = objFile.GetParentFolderName(strFile)
    If Len(strFilePath) = 0 Then strFilePath = App.Path
    With cnAccess
        .ConnectionString = "DBQ=" & strFile & ";DefaultDir=" & strFilePath & _

```

```

";Driver={Microsoft Access Driver (*.mdb)};uid=admin;pwd=scsl"
.Open
If .State = adStateClosed Then Exit Function
strSQL = "Select A.ReqNO as 编号,A.ExeDate as 日期, _
        A.GermID as 细菌,B.samNo as 标本号 "
strSQL = strSQL & " From ResultExe A,ResultBill B "
strSQL = strSQL & " Where A.ReqNO=B.ReqNO _
        And Format(A.ExeDate,'yyyy-MM-dd')>=" & _
        Format(dtStart, "yyyy-MM-dd") & ""
strSQL = strSQL & " And Format(A.ExeDate,'yyyy-MM-dd')<=" & _
        Format(dtEnd, "yyyy-MM-dd") & "" & _
IIf(IngSampleNo = -1, "", " And val(A.ReqNo)=" & IngSampleNo)
‘上面这一句处理传入的参数，如果是按标本号返回结果，则加上标本号条件。
Set rsTmp = cnAccess.Execute(strSQL)
If rsTmp.EOF Then
    .Close
    Exit Function
End If

IngResultCount = 0
ReDim aResults(IngResultCount)
strBh = ""
Do Until rsTmp.EOF
    ‘循环处理标本，标本号不同则当成新的检验结果。
    ‘因为同一个人可能有两个以上的细菌结果
    ‘所以细菌不同，也当成新的检验结果
    If strBh <> rsTmp("编号") Or str 细菌 <> Trim("" & rsTmp("细菌")) _
        Then

        ‘处理字符串数组，如果有内容就增加一个，没内容就用当前的
        IngResultCount = ubound(aResults)
        If aResults(IngResultCount)<>"" then
            IngResultCount= IngResultCount+1
            ReDim Preserve aResults(IngResultCount)
        End If
        ‘细菌结果部分
        aResults(IngResultCount) = _
            Format(rsTmp("日期"), "yyyy-mm-dd hh:mm:ss") '检验时间
        aResults(IngResultCount) = aResults(IngResultCount) & "|" & _
            Val(rsTmp("标本号")) '样本序号
        aResults(IngResultCount) = aResults(IngResultCount) & "|"
        aResults(IngResultCount) = aResults(IngResultCount) & "|" & _
            IIf(IsNull(rsTmp("细菌")), " ", rsTmp("细菌")) '标本
        aResults(IngResultCount) = aResults(IngResultCount) & "|0"
    End If
    strBh = rsTmp("编号")
    str 细菌 = rsTmp("细菌")
    rsTmp.MoveNext
Loop

```

```

str 细菌 = Trim("'" & rsTmp("细菌"))
strBh = rsTmp("编号")
strAnti = ""
'抗生素结果部分
strSQL = "Select AntiID,MicExplain,MicResult,ResShow _
          From MicResult Where val(ReqNo)=" & strBh
Set rsAnti = cnAccess.Execute(strSQL)
Do Until rsAnti.EOF
    strAnti = strAnti & "|" & rsAnti!AntiID & "|" & _
              rsAnti!MicExplain & "^1^" & rsAnti!ResShow
    rsAnti.MoveNext
Loop
aResults(lngResultCount) = aResults(lngResultCount) & strAnti
End If
rsTmp.MoveNext
Loop
End With
ResultFromFile = aResults
End Function

```

至此，仪器接口的开发内容就介绍完毕，在开发仪器接口时，注意下面三点，可提高接口的效率和缩短调试周期：

- 1.接口中要有明确的开始结束判断方法。
- 2.接口中对收到的信息进行要判断处理后根据情况再应答，不要一收到信息就应答。
- 3.最好在接口中缓存接收到的数据。

## 4.5、接口调试详解

简单的说，接口调试就是用工具模拟仪器或者直接在仪器上向通讯程序发送和接收数据，通讯程序在这个过程中，会调用指定的仪器接口进行解码，如果接口程序逻辑正确，则会解出格式正确的结果，否则，就根据产生的日志，对比接口程序分析找错误原因，修改后再测，直到通过为止。前面提到，接口调试可用工具模拟或直接在仪器调试，但是如果本机模拟调试做得好的话，可以缩短实际环境下的上机调试时间，节约实施时间，减少医院仪器的试剂的损耗，所以，如果一个新接口，先在模拟仪器上调试好，再在真实环境下上机调试是比较好的流程。当然，也有些仪器因为安装有控制软件在电脑上，控制软件可以保存检验数据，可以直接在真实环境下任意多次发送数据，不会产生试剂损耗，这种仪器在调试时，如果机器空闲时间较多，就没有必要建立测试环境了。总之，是否用工具模拟仪器调试，现场人员可以根据情况灵活应用。

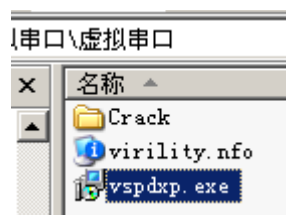
因实际环境下每个仪器的操作不尽相同，并且上机调试与模拟调试的结果差别不大，对于仪器上的操作大家可以查仪器的操作手册，这里我们主要介绍本机模拟调试的方法和工具。

我们在《怎样进行数据采集.doc》中提到，与仪器的连接，有网线与串口两种方式，分

别有不同的采集数据工具，实际上，采集数据的工具，我们也可以把它当成是调试仪器的工具，我们把采到的数据，用采集数据的工具向通讯程序发送，那么，这个采集工具就成了模拟仪器。

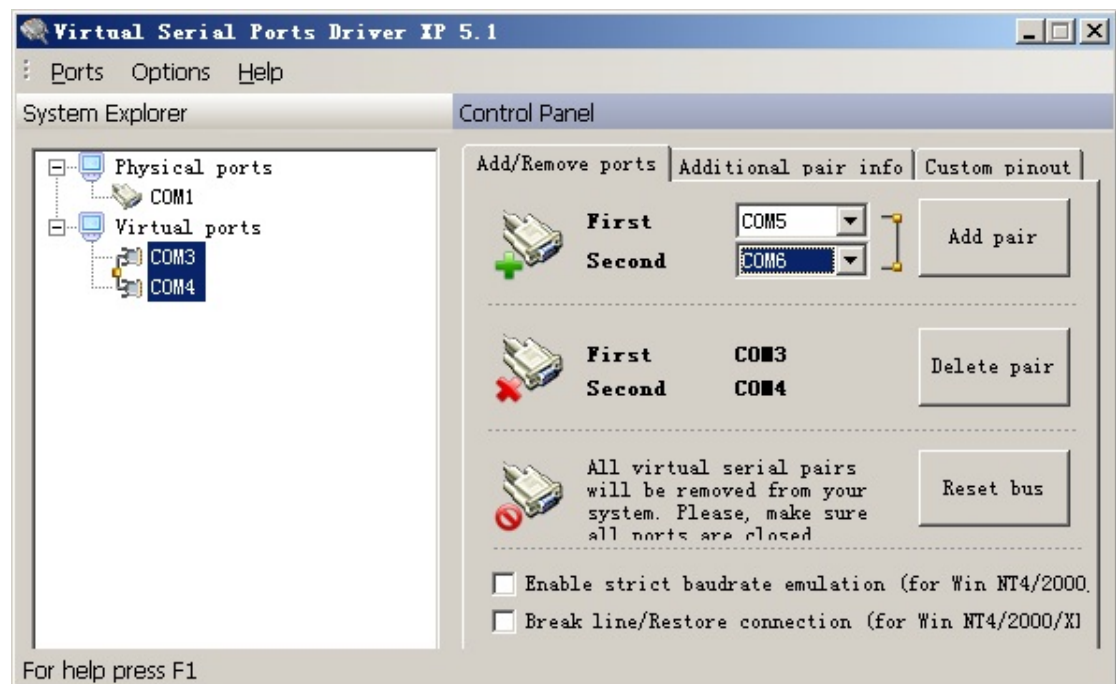
用串口通讯的仪器，现在大多数电脑只有一个串口，正常情况下，用一台电脑不能完成测试，需要两台电脑用数据线连接。我们找到一个虚拟串口软件，可以在本机模拟多个串口，并在软件中将两个串口连接上，连制作数据线的工作都省了，对于没有串口的笔记本电脑来说，也能在本机进行串口通讯测试。下面简单介绍下这个软件：

首先安装这个软件，单击“vspd xp.exe”即可，如图：



安装完成后，把 Crack 目录下的破解文件拷到安装目录，替换同名文件即可。

运行 vspd xp，在主界面上点设定好 2 个串口之后，点“Add pair”即可，如下：



上图中的电脑上 COM1 是物理串口，未连接；COM3 和 COM4 是两个虚拟串口，并且已连接，用串口工具分别打开 COM3 和 COM4，就可以互相发送和接收数据了。

我们可以按以下步骤进行仪器接口调试

## 1.添加仪器



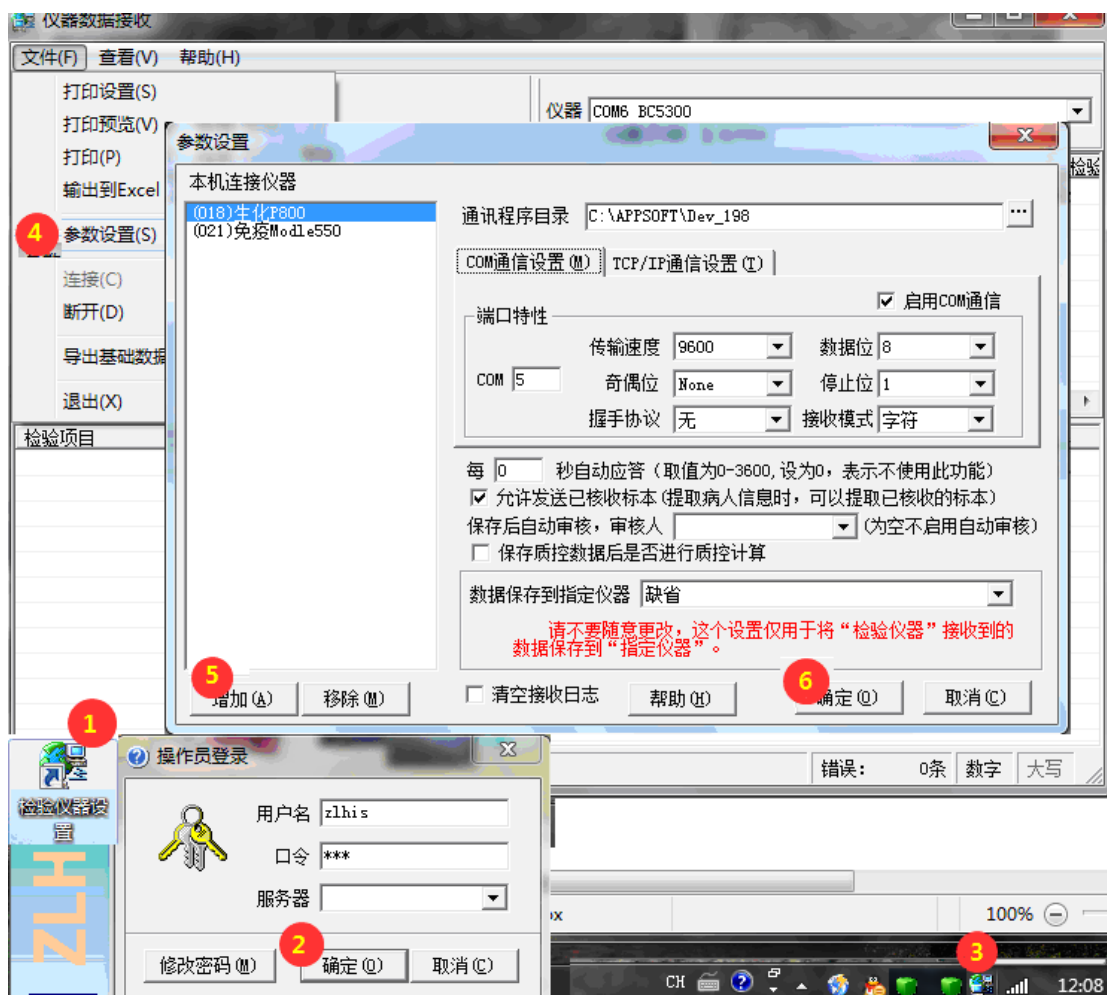
先增加要调试的仪器，指定仪器的通讯程序名称，保存后，再设置仪器的项目通道，把该仪器能做的项目添加进去，调试期间，可以不加全，加几个就可以了。通道码这里可以先随便填一个，后面我们可以从日志中查到。

注意，在修改了通道码后，通讯程序需要重新启动，才生效。

## 2. 设置通讯参数

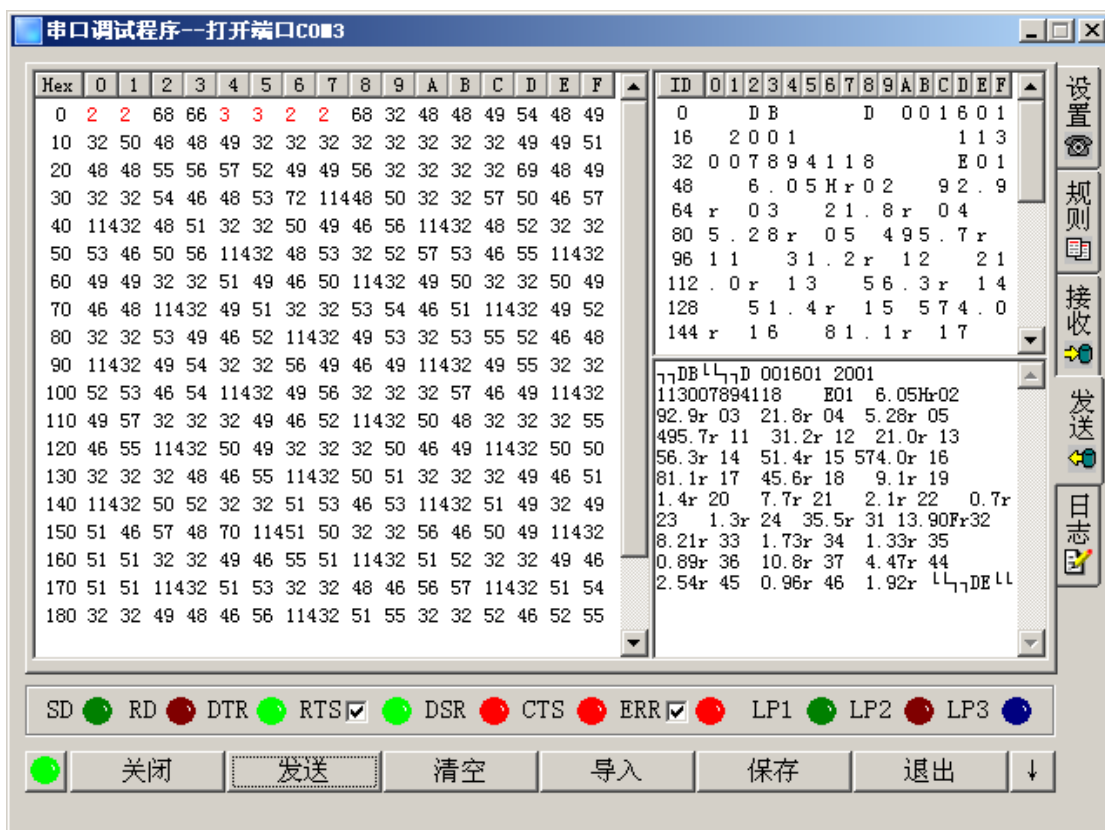
按下图中的步骤操作，设置仪器的通讯参数，因在 10.25.0 中对通讯程序进行了处理，现在，和仪器通信的程序放在指定的目录下，每个仪器一个目录，现在启动后，可能会有几个绿色立方体的图标在状态栏显示，一个绿色立方体图标就代表一个通讯接口程序。





### 3.发送数据

用工具模拟仪器发送数据，比如 ComTest，如下：

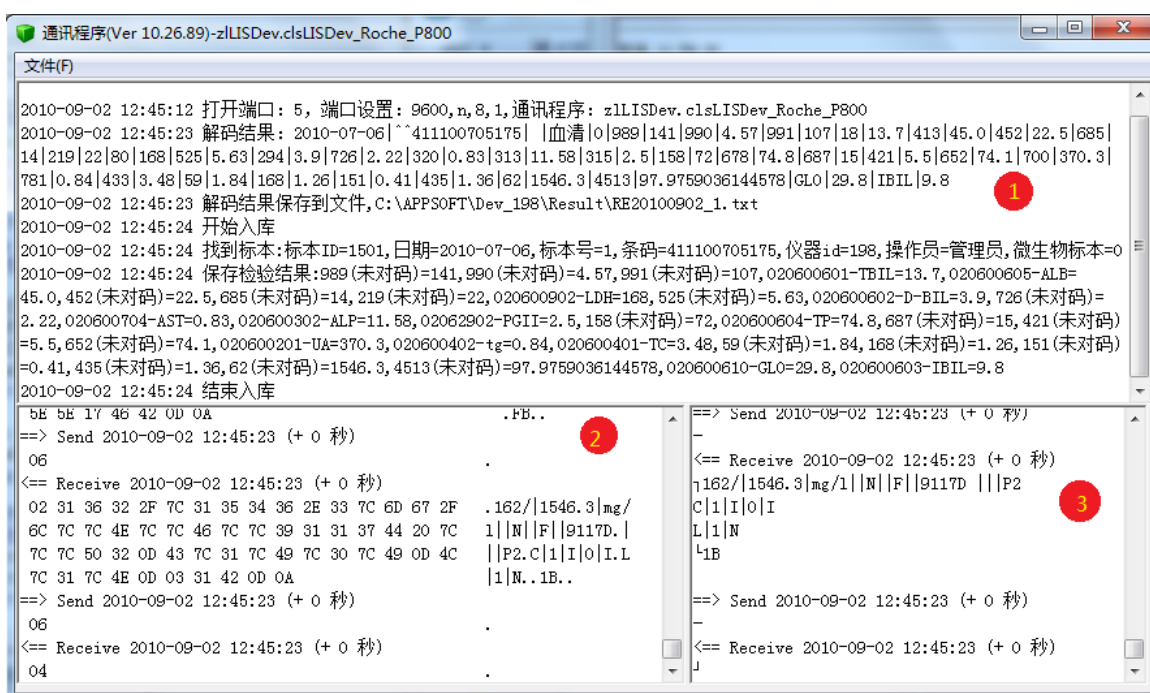


如果要在本机调试双向仪器，用这个工具就不太方便了，大家可以用其他调试工具来发送数据。

如果条件允许，也可以直接在仪器上发送数据，这样调试好了之后就可以直接使用了。

#### 4. 调试时生成的日志

10.27.0 发布的通讯程序中，当选择了“文件-调试”后，即可显示调试区域。如下图：



上面窗体中，区域 1 就是调试信息显示区域，要显示这个区域，需要在选择菜单“文件

-调试”，当完成一个关键步骤后，就会在此窗体中显示此步骤的执行情况，如果在技师工作站中没有收到结果，我们就可以根据窗体中显示的信息定位故障点，缩短调试时间。区域 2 是以十六进制显示收发数据的详细信息，这个区域中显示的信息模仿 **HHD** 的格式；区域 3 以字符形式显示收发数据的讯详细信息，通过这两个区域，我们可以得到通讯过程中接收和发送的详细情况，方便事后查问题。在调试模式下，当 1，2，3 区域中显示的信息超过 1000 行时，会自动将已显示的信息保存为一个文件，文件在程序所在目录下，如果有通讯问题，那么这个日志文件可以供当时不在现场的技术人员分析。