

OpenCPN 探索—绘图指令解读

概述：

前面已经发表过 OpenCPN 海图显示的相关文章，海图显示一文主要从函数级别上讲解显示的流程步骤。今天的绘图指令解读主要从文件级别上解析绘图指令含义以及在 OpenCPN 上是如何分解转义该指令的。

OpenCPN 中绘图指令初始化获取的方式有两种：

- 1: 根据 obj 对象查找对应的 S52 绘图指令；
- 2: 条件符号化的绘图指令转移到回调函数中动态生成；

具体生成指令的方式将在后面发表的文章中详述，本文先针对 S52 中的绘图指令进行分析。

分析：

OpenCPN 中 S52 绘图指令主要来源于 chartsymbols.xml/S52RAZDS.RLE,这两个文件可以在 OpenCPN 工程中找到。两个文件格式不太一样，但是内容几乎是一样的，OpenCPN 中对这两个文件进行了区分，S52RAZDS.RLE 属于旧格式文件，chartsymbols.xml 是新格式文件。为了方便阅读，本文以 chartsymbols.xml 作为介绍对象。

chartsymbols.xml 介绍

文件结构：文件分为 5 部分，分别是：

- 1: color-tables 记录的是 S52 标准下各种现实模式的基准色；
- 2: lookups 记录的是各种类型 S52 标准的点线面绘制；
- 3: line-styles 记录的是各种线型；
- 4: patterns 记录的是各种类型图案的绘制；
- 5: symbols 记录的是各种类型符号的绘制；

这几部分的关系是：lookups 提供总体绘制方法，在绘制方法中调用其他部分的定义实现最终的绘制。也就是说，一个 obj 对象的绘制，首先需要定位到它属于 lookups 中的哪一类，再根据 lookups 的记录查找线、图案、符号和颜色完成绘制和填充。

在 OpenCPN 中 obj 对象绘制前必须与 S52 文件中的一个 lookups 匹配上，根据匹配的结果中记录的绘制指令进行绘制。

在文件中各个符号简称是有意义的：

SY-----符号类型，它带的参数需要到 symbols 中去找对应的绘制指令；
CS-----条件符号化，这个目前比较复杂，在 OpenCPN 中很多条件符号化的指令都是在代码中动态生成的，本文对此不做介绍，以后针对条件符号化解析会有详细介绍。

LS-----简单线，在 color-tables 查找颜色，宽度和线型在括号中；

LC-----复杂线，在 symbols 中可以查找指令；

AC-----颜色填充，在 color-tables 中查找颜色；

AP-----图案填充，在 patterns 中查找图案

TE-----数值类文字，需要字母加以区分

TX-----普通字符串

OpenCPN 工程根据这些将 obj 进行匹配，最终得到一串绘制指令，类似于：
SPC;PU1500,565;ST0;PM0;CI110;PM2;FP;PU1500,840;PM0;CI110;PM2;FP...这个指令即用来完成最终图像绘制。解读这个指令的函数是 RenderFromHPGL::Render。

解析函数介绍

解析函数 RenderFromHPGL::Render 的理解最好能跟着一段解析调试过程进行理解。

解析的步骤：

- 1: SP-----获取绘制的画笔颜色和画刷颜色；
- 2: SW-----获取画笔的宽度；
- 3: PU
A:设置画笔
B:获取线的起始点
C:偏转计算
- 4: PD
A:获取线的结束点
B:绘制一根线
- 5: CI-----设置半径直接画圆；
- 6: PM-----绘制复杂多边形，中间会调用 PD、CI 绘制线和圆；
- 7: FP-----绘制多边形；

指令解释（请参考 S52 标准）：

SP 用于选择一种画笔颜色，参数是一个字母（ASCII>=64），该参数标示一种 S52 颜色表中标定的颜色。当一种画笔颜色被选定以后直到出现新的 SP 指令选择新的画笔前，所有的绘制指令均采用上一次设置的画笔进行绘制。

ST 透明度，仅对填充区域有效。

SW 宽度，定义了线的物理宽度。

PU 移动的过程，中间并没有任何绘制过程，仅仅是移动画笔。

PD 绘制的过程，移动过程中完成一条线的绘制。

CI 绘制圆的过程，绘制完成后，画笔重新回到圆心。

AA 绘制弧的过程, 顺时针绘制, 绘制完成后不返回画笔初始位置。

PM 多边形定义模式。在该模式下, 您可以使用其他指示构造多边形 (PU、PD、CI、AA)。这些指令都在存储缓冲区多边形; 他们不执行, 直到多边形完全定义。为了绘制多边形必须填补它与 FP 指令和/或 EP 指令。

EP 绘制已经存储的多边形。完成后, 恢复原来的笔的位置和状态。

FP 填充已经存储的多边形。完成后, 恢复原来的笔的位置和状态。

SC 调用一个符号定义, 完成符号的绘制。