

# 二维条码 PDF417编码原理及其软件实现

赵博, 黄进  
(西安电子科技大学, 西安 710071)

**摘要:** 介绍了二维条码 PDF417的编码原理, 给出了在文本压缩模式下编码的程序流程图, 并且用 Visual Basic语言 完成了其编码打印软件, 经实验验证所编程序可以准确无误地实现 PDF417条码的编码和打印。

**关键词:** PDF417条码; 编码算法; 条码打印软件

**中图分类号:** TP391. 44 **文献标识码:** A **文章编号:** 1001- 3563( 2007) 07- 0096- 03

## Coding Principle of Two Dimension Bar Code PDF417 and Its Software Implementation

ZHAO Bo HUANG Jin  
(Xi'an University, Xi'an 710071, China)

**Abstract** The principle of coding two dimension bar code PDF417 was introduced. The flow chart of coding bar code under text compression mode was provided, and the software of coding and printing was written with Visual Basic. It was proved that the program can execute the coding and printing function correctly.

**Key words** PDF417 bar code; arithmetic of coding; bar code printing software

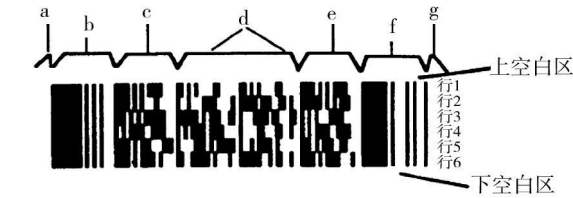
二维条码技术是在计算机与信息技术基础上发展起来的一种标准化信息存储、传递和自动识别技术, 具有成本低、容量大、可靠性高、编码方式灵活、保密防伪性强、不依赖后台数据库等优点。其中应用最广泛的是美国 Symbol公司发明的 PDF417二维条码, 在国外已被使用在国防、公共安全、交通运输、物流、医疗保健、工业自动化、商业、金融、海关及政府管理等领域, 并取得了巨大的社会效益和经济效益<sup>[1- 2]</sup>。

在我国, 大部分的技术和设备目前仍然以引进居多, 致使二维条码使用成本较高, 限制了二维条码的普及应用, 通过 Visual Basic 6. 0编制实现二维条码编码打印软件, 为二维条码 PDF417在我国的进一步普及应用作了一些有益的工作。

### 1 PDF417条码简介

PDF是取英文 Portable Data File 3个单词的首字母缩写, 意为“便携数据文件”。因为组成条码的每一符号字符都是由4个条和4个空构成, 如果将组成条码的最窄条或空称为一个模块, 则上述的4个条和4个空的总模块数为17, 所以称四一七条码或 PDF417码<sup>[3- 4]</sup>。条码的符号结构见图1。

PDF417条码的符号尺寸是可变的, 高度范围为390行, 宽度范围为90583X, X是符号的模块宽度。条码的最大数据量



a—左空白区; b—起始符; c—左行指示符号字符; d—130个数据符号字符; e—右行指示符号字符; f—终止符; g—右空白区。

图1 PDF417条码的符号结构

Fig. 1 PDF417 barcode structure

(错误纠正等级为0时)为每个符号表示1850个大写字母或2710个数字或1108个字节。

### 2 二维条码 PDF417的编码

PDF417条码的编码分为数据码字编码、错误纠正码字编码、前后行指示符编码3个部分。有3种数据压缩模式: 文本压缩模式(TC); 字节压缩模式(BC); 数字压缩模式(NC)。

#### 2.1 字节压缩模式(BC)

字节压缩模式通过基256至基900的转换, 将字节序列转换为码字序列。

收稿日期: 2006-09-11

基金项目: 教育部留学回国基金; 喷墨式CTP关键技术项目资助(K40604040114)

作者简介: 赵博(1980-), 男, 西安人, 西安电子科技大学, 硕士研究生, 主要研究方向为精密测试技术等。

对于字节压缩模式,有 2 个模式锁定 (901, 924): (1) 当所要表示的字节总数不是 6 的倍数时,用模式锁定 901; 当所要表示的字节总数为 6 的倍数时,用模式锁定 924。在应用模式锁定 924 的情况下,6 个字节可通过基 256 至基 900 的转换用 5 个码字表示,从左到右进行转换。(2) 当所要表示的字节数不是 6 的倍数时,必须使用模式锁定码字 901。前每 6 个字节的转换方法与上述方法相同,对被 6 整除所剩余的字节应每个字节对应一个码字,逐字节用码字表示。

2.2 数字压缩模式 (NC)

数字压缩模式是指从基 10 至 900 的数据压缩的一种方法。数字压缩模式能把约 3 个数字位用一个码字表示。尽管在任意数字长度下都可以应用数字压缩模式,一般推荐当连续的数字位数大于 13 时,用数字压缩模式,否则应用文本压缩模式。

在数字模式下,将根据下述算法对数字位进行编码: (1) 将数字序列从左向右每 44 位分为一组,最后一组包含的数字位可以少于 44 个。(2) 对于每一组数字: 首先在数字序列前加一位有效数字 1 (即前导位),然后执行基 10 至基 900 的转换。

2.3 文本压缩模式 (TC)

以最常用的文本压缩模式为例来说明其计算机编码算法:

2.3.1 数据码字编码

子模式: 文本压缩模式是每一符号起始的默认有效的压缩模式。为了更有效的表示数据,文本压缩又分为 4 个子模式: 大写字母型子模式、小写字母型子模式、混合型子模式、标点型子模式。在子模式中,每一个字符对应一个值 (029)。

子模式之间的切换: 在文本压缩模式中,每一个码字用 2 个基为 30 的值表示 (范围为 029)。如果在一个字符串的尾部有奇数个基为 30 的值,需要用值为 29 的虚拟字符  $ps$  填充最后一个码字。这样就可以用一个单独的码字表示一个字符对,表示字符对的码字由下式计算:

码字 = 30×H + L

式中:  $H, L$  依次表示字符对中的高位和低位字符值。

2.3.2 错误纠正码

对于一组给定的数据码字,错误纠正等级根据 Reed Solom on 错误控制码算法计算。

纠正码字的计算步骤:

第一步: 建立符号数据多项式。

符号数据多项式如下:

$$d(x) = d_{n-1} x^{n-1} + d_{n-2} x^{n-2} + \dots + d_1 x + d_0$$

式中: 多项式的系数由数据区码字组成。其中包括长度码、数据码字、填充码和宏四一七条码控制块。其中  $d_n$  为数据码字  $d_0 \dots d_{n-1}$ ,  $n$  为数据码字数 (包括数据长度码字)

每一数据码字 ( $d_i, i = 0 \dots n-2, n-1$ ) 在 417 条码符号中的排列位置,见图 2。

起 始 符	$L_0$	$d_{n-1}$	$d_{n-2}$				$R_0$	终 止 符
	$L_1$						$R_1$	
	$L_{m-2}$		$d_0$	$c_{k-1}$	$c_{k-2}$	$R_{m-2}$		
	$L_{m-1}$				$c_1$	$c_0$	$R_{m-1}$	

图 2 数据、行标识符及错误纠正码的排列位置

Fig 2 The position of data row identifier and error correcting code

第二步: 建立纠正码字的生成多项式。

$k$  个错误纠正码字的生成多项式如下:

$$g_k(x) = (x - 3)(x - 3^2) \dots (x - 3^k) = \alpha_0 + \alpha_1 x + \dots + \alpha_{k-1} x^{k-1} + x^k$$

式中:  $k$  为错误纠正码字  $c_i (i = 0 \dots k-2, k-1)$  的个数;  $k = 2^{s+1}$ ,  $s$  为错误纠正等级;  $\alpha_k$  为多项式  $g_k(x)$  展开后各项的系数序列;  $c_i$  在 417 条码符号中的排列位置见图 2。

第三步: 错误纠正码字计算。对一组给定的数据码字和选定的错误纠正等级,错误纠正码字为符号数据多项式  $d(x)$  乘以  $x^k$ , 然后除以生成多项式  $g(x)$ , 所得余式的各系数的补数。如果  $c_i > -929$  在有限域  $GF(929)$  中的负值等于该值的补数; 如果  $c_i \leq -929$  在有限域  $GF(929)$  中的负值等于余数 ( $c_i / 929$ ) 的补数。

2.3.3 前后行指示符

行指示符号字符包括左行指示符号字符 ( $L_i$ ) 和右行指示符号字符 ( $R_i$ ), 分别与起始符和终止符相邻, 见表 1。行指示符号字符的值 (码字) 指示四一七条码的行号 ( $i$ ), 行数 (390), 数据区中的数据符号的列数 (130), 错误纠正等级 (08)。

左行指示符号字符 ( $L_i$ ) 的值由下式确定:

$$L_i = \begin{cases} 30x_i + y & \text{当 } c_i = 0 \\ 30x_i + z & \text{当 } c_i = 3 \\ 30x_i + v & \text{当 } c_i = 6 \end{cases}$$

右行指示符号字符 ( $R_i$ ) 的值由下式确定:

$$R_i = \begin{cases} 30x_i + v & \text{当 } c_i = 0 \\ 30x_i + y & \text{当 } c_i = 3 \\ 30x_i + z & \text{当 } c_i = 6 \end{cases}$$

式中:  $x_i = \text{NT}[(\text{行号} - 1) / 3], i = 1, 2, 3, \dots, 90$

$y = \text{NT}[(\text{行数} - 1) / 3]$

$z = \text{错误纠正等级} \times 3 + (\text{行数} - 1) \bmod 3$

$v = \text{数据区的列数} - 1$

$c_i = \text{第 } i \text{ 行簇号}$

3 在 Visual Basic<sup>[5]</sup> 下实现编码软件

3.1 PDF417条码在文本模式下编码程序

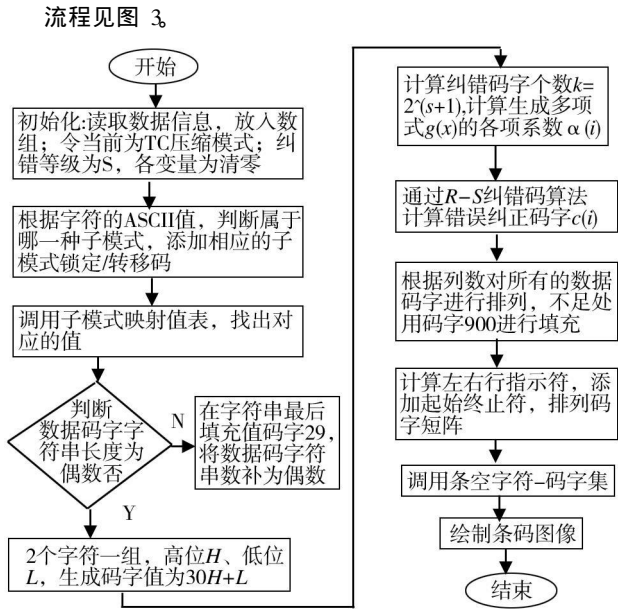


图 3 编码程序流程图

Fig 3 Flow chart of coding program

3.2 PDF417条码生成程序友好界面

见图 4所示友好界面中,从便于研究验证的角度出发,设置了生成、显示数据码字和条空字符的数据,以便每一步都可以监控到数据变化的正确与否,从而通过分析数据为发现问题,调试程序提供直接依据。

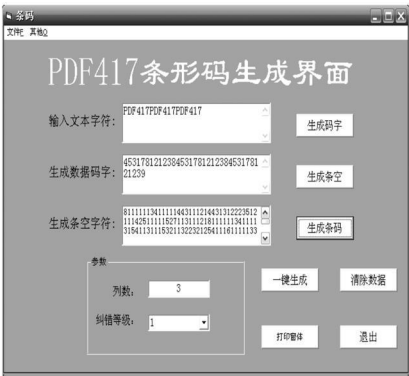


图 4 PDF417 条码生成程序友好界面

Fig 4 Friendly interface of PDF417 barcode generation software

当输入文本字符、列数(这里指数据码字的列数)、选择纠错等级后,直接点击一键生成按钮,就可以直接从显示窗口 Form2上显示出相应的 PDF417 条码,然后点击打印窗口按钮,就可以直接打印出对应条码,简便明了。当然我们可以将 PDF 417 条码保存为 mdi 格式的图像,然后按照实际需求来调

整图像的长宽比、大小等,之后打印出合适的条码图像。这里给出了常用的文本压缩模式下的程序流程图,其它 2 种模式下编码情况根据本文开始所列的各自算法来完成,过程与文本压缩模式下的条码生成相类似。

3.3 生成 PDF417 二维条码

条码生成程序从第一行开始分黑色和白色模块按条空字符数据画出小矩形条,从左至右,从上到下,逐行绘制,直到最后一行结束,见图 5。

条码原始信息: PDF417PDF417PDF417



图 5 实验中软件生成的条码

Fig 5 Barcode generated by the software

4 结 语

本文实现的 PDF417 编码系统具有信息录用方便、易于操作维护、界面友好等优点。实验结果表明所生成的 PDF417 二维条码,可以被二维条码扫描仪准确无误地读取从而获得原始信息。此外,此 PDF417 编码系统可方便地移植到打印控制器中,从而方便的制成 PDF 417 二维条码打印机。