

Coloured Region Barcode Printing

Introduction

TPCL-based TOSHIBA TEC Barcode Printers have built-in support for printing many different kinds of barcodes, however, for general purpose printing from word processor applications, there is no simple method for the user to invoke such features in the barcode printer. This means that the user must rely on pre-generated barcodes if they want the printed label to contain a barcode.

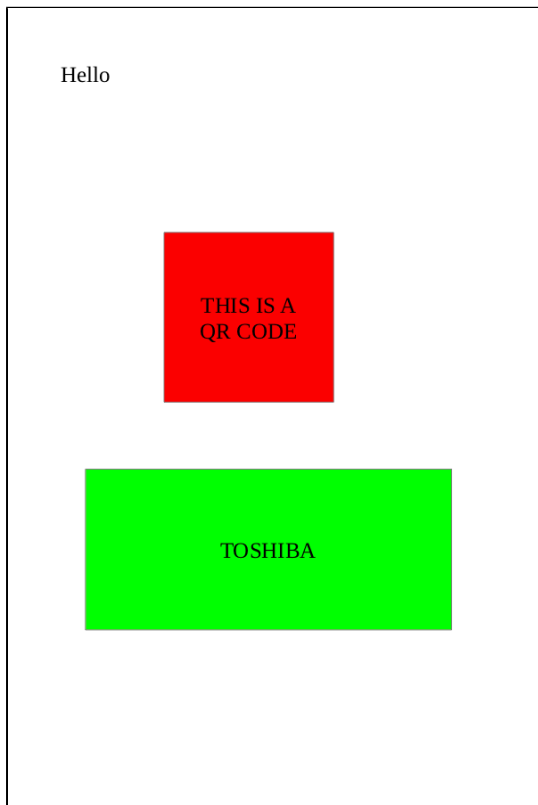
A simple method described here allows users of the TOSHIBA TEC Barcode Printer Drivers for Linux to put barcodes onto the printed label quickly and effectively, and it should be supported by virtually any application that either produces PDF documents natively or prints via a CUPS-compatible interface such as GTK/Cairo.

At the moment, only two types of barcodes are currently supported using this method, QR codes, and Code93, however, this concept can be extended to any barcode type supported by the printer.

Inserting a Barcode

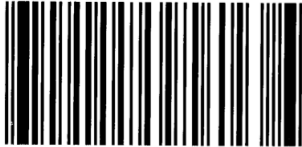
In order to place a barcode on the label, simply create a region on the document that has the colour RGB(0, 255, 0), or RGB(255, 0, 0). The colour value does not have to match exactly, there is a small tolerance of approximately ± 10 . At the moment these are the only supported colours, however, as mentioned prior, this can be extended.

To best illustrate how this works, below is a document containing both a red region RGB(255, 0, 0), and a green region RGB(0, 255, 0). This document was created in LibreOffice Write, however, any application that is capable of producing documents with coloured regions should be supported.



And here is the result when this document is printed using TOSHIBA TEC Barcode Printer Drivers for Linux:

Hello



The red square has been replaced with a QR code, and when it is scanned, it should contain the text "THIS IS A QR CODE". The green rectangle has been replaced with a Code93 barcode, and when this is scanned, it should contain the text "TOSHIBA".

The position, size and shape of the coloured region determines the position and size of the barcode when it is printed. The driver will determine the maximum size of the barcode that fits within the coloured region and issues the appropriate TPCL command to produce a barcode of that size using the specified value. Effectively the coloured region is a "bounding box".

The value of the barcode is determined by the text that appears in the coloured region.

Notes:

- Any number of coloured regions can appear on the page
- Overlapped coloured regions are not supported (the behaviour is undefined)
- The origin of the text (the coordinates of the first character) must be within the coloured region in order for the text to be used as the barcode value
- If the text is at 90°, the barcode will be placed at 90° (similarly, text that is rotated at 180° will cause the barcode to be printed at 180°, and text at 270° will cause the barcode to be printed at 270°).
- Text that is not at a fixed rotation of 0°, 90°, 180°, or 270° will be **discarded**
- If the shape does not consist of exact horizontal and vertical lines, it will be printed as a filled shape instead. For example, the four points that make up a rectangle starting at top-left going clockwise: (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , (x_4, y_4) , the following must hold true:
 - $x_1 = x_4$
 - $x_2 = x_3$
 - $y_1 = y_2$
 - $y_3 = y_4$
- In the above calculation, if there are any superfluous collinear points, they are automatically stripped.