



PIPSTA006 – PIPSTA QR-CODES

Contents

Difficulty Level:.....	2
Time to Complete:	2
Who Should Read This Document	2
Pre-Requisites	3
Getting Started.....	3
How it Works.....	4
The QR Script Command Line	4
Changing the Text	4
Next Steps	5
Shutting Pipsta Down Safely	5

Revision History

Revision	Author	Date	Description
1.0	AH	22/11/14	First Release

Difficulty Level:



- No Linux or Raspberry Pi knowledge is required

Time to Complete:



- This tutorial is limited to demonstrating the script, rather than explaining how it works

Who Should Read This Document

Pipsta is capable of printing QR-Codes (2D barcodes) to either paper or linerless labels for subsequent decoding by commonly available and free smart phone apps.

This guide is suitable for users wishing to experiment with QR-Codes, or those wishing to incorporate QR-Codes into their displays, advertisements or other materials.



Pre-Requisites

This guide is intended for users who have completed the mechanical build of their Pipsta and have successfully performed the first-time setup concluding in a simple print-out, as per:

- **PIPSTA002 - Pipsta B+ Assembly Instructions** or
- **PIPSTA003 - Pipsta A+ Assembly Instructions**
and
- **PIPSTA004 – Pipsta First-Time Setup**

It is expected that you have the following items:

- Your assembled, working Pipsta
- Power supply for Raspberry Pi (5v, ideally no less than 2.0A rated)
- USB A to Micro B cable assembly (if not integrated into Raspberry Pi power supply)
- Micro SD Card configured as per the *Pipsta First-Time Setup tutorial*
- Consider an additional, reserve Micro SD Card!
- Access to two mains sockets for powering the Raspberry Pi and Pipsta printer
- USB Keyboard
- USB Mouse
- A Wi-Fi connection as per the *Pipsta First-Time Setup tutorial*
- Video/Monitor lead:
 - HDMI lead *or*
 - 3.5mm 4 pole jack plug to RCA composite cable *or*
 - HDMI to VGA adaptor and VGA cable
- Computer monitor or television (with HDMI, component video or VGA input as above)
- Smart phone with QR-Code reader app

Getting Started

- 1) Power up your Raspberry Pi and printer
- 2) Wait for the Raspberry Pi to boot into the graphical desktop environment
- 3) Click on *File Manager* on the task-bar.
- 4) Navigate to **/home/pi/pipsta/7_QR_Print**
- 5) Press **[F4]** to open LXTerminal
- 6) At the \$ prompt, enter:
python qr.py
- 7) The Raspberry Pi will instruct the printer to flash its green LED to signal that the QR-Code image encoding is in progress. The QR-Code is optimised for the optimal resolution for the Pipsta printer, and –as QR-Codes are square—the Raspberry Pi image processing delay will generally be far less than –say—the delay for producing long banner prints.
- 8) Once the encoding is complete, a default QR-Code will be printed.
- 9) Ensuring that the paper/label is flat, scan the QR-Code with the smart-phone app to read its contents.





TIP:

For those with little programming experience, if your smart-phone app displays text interspersed with `\r\n`, this is a common sequence at the end of lines of text:

- `\r` is an *escape code* for carriage return
- `\n` is an *escape code* for newline

The typical result is that the next characters displayed will be on a new line, and the position at which characters will appear will have returned to the left-hand side of the display. Examine the file 'qr.txt' to see how this was encoded.

How it Works

- 1) In this simple demonstration mode, the Python script running on the Raspberry Pi reads the content of the file "qr.txt"
- 2) The text is used to generate a QR-Code bitmap image of the contents of the file in the Raspberry Pi's RAM, using the Python Imaging Library, Pillow.
- 3) The bitmap image is converted to a suitable format before then being sent over USB to the printer as a series of graphics commands. The bytearray library is used to convert the image to blocks of data, and PyUSB and LibUSB are used to send the data to the printer.

The QR Script Command Line

The QR script takes a single *command-line argument* of the file (and its path). This file contains the text to be printed.

Note that—in order to make the above demonstration as simple as possible—the argument is optional.

Changing the Text

In order to demonstrate the command line option, try creating a text file as follows:

- 1) In File Explorer, navigate to **/home/pi/Pipsta/7_QR_Print**
- 2) Press **[F4]** to bring up LXTerminal
- 3) At the \$ prompt, enter:

nano test.txt

...to bring up the in-built editor

- 4) Type:
Another test of Pipsta QR-Codes
- 5) Press **[CTRL]** and **[X]** to exit
- 6) Press **[Y]** to save
- 7) Press **[ENTER]** to accept the existing filename
- 8) In LXTerminal, at the \$ prompt, now enter the following command:

python qr.py test.txt

A short time later a QR-Code will be produced. Scan this QR-Code with your smart-phone app to confirm its contents.



Next Steps

If this is the end of your session, see the section entitled *Shutting Pipsta Down Safely*, or –to continue on to more advanced applications— take the next step with any of the following tutorials:

- ***PIPSTA005 – Pipsta Banners***
- ***PIPSTA106 – QR-Code Python Code Tutorial***

Shutting Pipsta Down Safely

Whilst the printer is resilient when it comes to powering down, the Raspberry Pi must undergo a strict shutdown process to avoid corrupting the Micro SD card. The most straightforward method of doing this is to double-click the ‘Shutdown’ icon on the desktop.



TIP:

If you are already in LXTerminal, type **sudo shutdown -h now** to shut-down the Raspberry Pi immediately.



TIP:

Always make sure ALL activity on the Raspberry Pi’s green LED (the LED on the right) has stopped before removing the power!

■End of Document■