PIPSTA005 – PIPSTA BANNERS

Contents

Revision History

Revision	Author	Date	Description
1.0	AH/GJ	27/11/14	First Release

Difficulty Level:

- No Linux or Raspberry Pi knowledge is required
- Although some technical terms are provided for developers, such aspects can be safely ignored without detriment to later stages of the tutorial.

Time to Complete:

- This tutorial is limited to demonstrating the app, rather than explaining how it works
- Time for completion will vary depending on length of banner messages
- Research into available fonts will obviously impact completion time

Who Should Read This Document

Pipsta is capable of printing large banners in a variety of fonts, to either paper or linerless labels. Pipsta banners can produce striking, high-quality headings and titles for classroom displays and high-visibility, eye-catching labels in a matter of minutes. Whether using solid fonts for instant results, or hollow fonts for subsequent colouring, Pipsta banners can draw upon an almost inexhaustible range of free or paid-for online fonts to ensure every banner produced is distinctive.

This guide is suitable for those solely wishing to use the script to produce banners, but it is also useful as an introduction to banner printing functionality for developers.

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 Ap1400 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)

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/6_Banner_Print**
5) Press **[F4]** to open LXTerminal
6) At the $ prompt, enter:
   python banner.py
7) The Raspberry Pi will instruct the printer to flash its green LED to signal that the font encoding is in progress. The Raspberry Pi is small and cheap, but it is not a powerful computer, so this may take some time!
8) Once the encoding is complete, a default banner will be printed.

How it Works

1) In this simple demonstration mode, the Python script running on the Raspberry Pi produces a default message of "My First Pipsta Banner" in a pre-installed font.
2) The text and font are used to generate a bitmap image in the Raspberry Pi's RAM, using the Python Imaging Library, Pillow.
3) This image is then rotated so it is oriented correctly to be printed (again, using Pillow).
4) 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 *bitarray* 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 Banner Script Command Line

The banner script takes two *command-line arguments*:

1) The text to be printed (in quotes),
2) The filename (and path) of the font to be used.

Note that the arguments are *positional*, meaning that the first string **must** be the message, and the second string must be the font file path.

Also note that –in order to make the above demonstration as simple as possible – both arguments are optional, meaning that *either* or *both* arguments may be omitted. It should be noted that attempting to omit the text and supply the font file will not be successful on account of the positional requirements.

Changing the Text

In LXTerminal, at the $ prompt, now enter the following command:

python banner.py "Test"

A short time later a new banner message will be produced. You ought to notice that the banner is produced significantly more quickly; this is due to the shorter message producing a smaller bitmap: requiring less resource and therefore time to manipulate.

TIP:

This basic banner script works in this way for ease of understanding, but there are many options for speeding this process up, including:

1) Breaking the image up into smaller pieces and printing each 'chunk' during the encoding of the next part,
2) Implementing the functionality in a *compiled language*, such as C or C++, rather than the *interpreted language*, Python.
3) Adjusting the overclocking settings of the Raspberry Pi. **Note that doing so may cause system instability and/or shorten the life of your Raspberry Pi!**

NB: As the text is supplied as a *string* in the command line, it is not possible to use characters such as double-quotes, carriage returns etc.

A small modification to this script can be made to provide a filename containing the message to be printed instead of the message itself. This would then permit the use of additional special characters, such as double-quotes.

It is possible to create very long banners in this way, however, the memory available to the Raspberry Pi to generate, store and manipulate the bitmap image may well be the limiting factor in just how long a banner may be.

Changing the Font

In order to change the font, you must know the path and filename of alternative fonts to use. There are many pre-installed fonts available in the Raspbian operating system distribution; these can be explored by using the following command at the $ prompt in LXTerminal:

```
fc-list
```

Now enter the following at the $ prompt:

python banner.py "More Fonts" "/usr/share/fonts/type1/gsfonts/z003034l.pfb"

Don't forget that you can use the [TAB] key to auto-complete folder names as you type and save yourself time and reduce typos!

It can be seen that the resultant banner uses a very different font.

You can save time and research what a font looks like by opening the LeafPad text editor and going to Options>Fonts. When you have found a font that you like (say –for example—the Chancery font used above), enter:
fc-list :fontFormat=OpenType | grep Chancery
(note case sensitivity!)
This will 'pipe' the full list of fonts into a search for the string 'Chancery'. This presents only those fonts that contain the string 'Chancery' and therefore allows quick identification of candidate fonts.

In addition to those to be found on the Raspberry Pi, there are many 'free for commercial use' fonts to be found on the web. This is a good way of finding fonts that have greater visual impact, and also identifying 'hollow' fonts that can be subsequently coloured with crayon or felt-tip.

Use a search engine to look for 'free for commercial use fonts'.
Once you have found a font e.g. '*burnstown_dam.ttf*', save it to e.g.:
/home/pi
…and then use LXTerminal **[F4]** from there to enter:
sudo cp burnstown.ttf /usr/share/fonts/
to copy the font file to the system font folder. You should then be able to check the font is present using:
fc-list | grep burnstown
…and inspect it using Leafpad Options>Fonts. Note that you may need to restart Leafpad before it will become visible in the font list.

Always check the terms of the licence against your usage.

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:

- ***PIPSTA006 – Pipsta QR Codes***

Or investigate how the code works by first reading:

- ***PIPSTA104 – Basic Print Python Code Tutorial***

Before then progressing to:

- ***PIPSTA105 – Banner Print 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.

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

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■