# PIPSTA108 –PIPSTA NFC PYTHON CODE TUTORIAL

## Contents

Revision History

| Revision | Author | Date | Description |
|---|---|---|---|
| 1.0 | AH | 26/11/14 | First Release |

## Difficulty Level: ●●●●●

- ➢ The NFC system spans multiple hosts and files, runs as a *daemon* and uses Linux, Python and Regular Expressions that are likely to be unfamiliar to newcomers

## Time to Complete: ●●●●○

- ➢ Many aspects are complex and require significant time to explain and comprehend.

## Who Should Read This Document

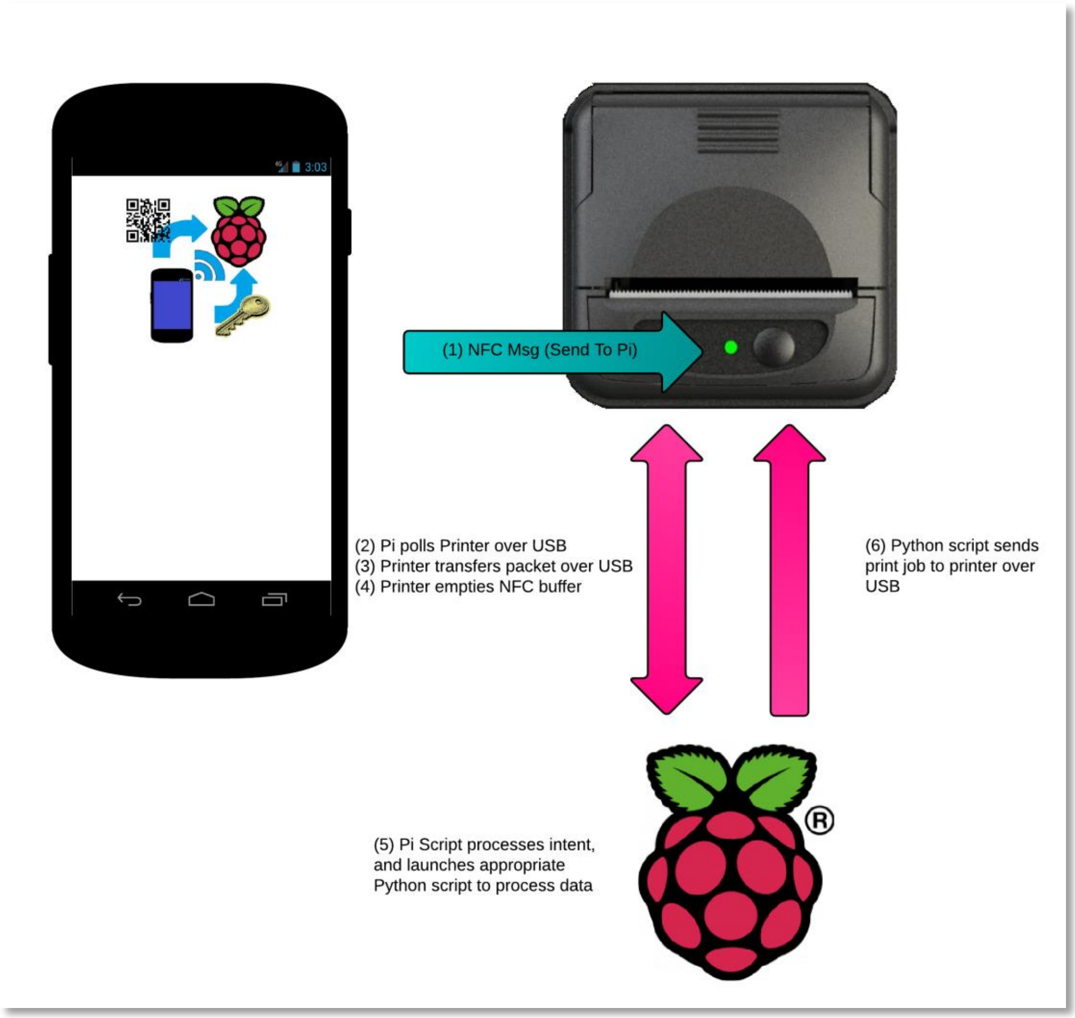
The complexity of this system means that this document is likely to be beyond the capabilities of beginners in Linux and Python programming. It may, however, be possible for some intermediate users to draw-upon and modify pre-existing techniques.

## Prerequisites

It is expected that your Pipsta is fully set-up and configured, that you have completed all aspects of the ***PIPSTA008 – Using NFC with Pipsta*** tutorial.

## How it Works

The NFC server system architecture is as follows:


3:03
(1) NFC Msg (Send To Pi)
(2) Pi polls Printer over USB
(3) Printer transfers packet over USB
(4) Printer empties NFC buffer
(6) Python script sends print job to printer over USB
(5) Pi Script processes intent, and launches appropriate Python script to process data

## NFC Message (Send to Pi)

In send to Pi mode, the Pipsta app sends a packet of NFC data over to the Pipsta printer. This packet contains:

1) Encoding (one of two text types)
2) Header: 0x01, 0x02
3) Method (e.g. pipsta.qr)
4) Data from the text field

In the current implementation, **the system is limited to a total of 64 bytes.** It is anticipated that this will be improved in future. In practice, keeping the text field data below 30 characters should suffice, but –if longer payload data is provided—the system will simply truncate this.

The printer decodes the header, and determines that the packet is intended to be returned to the Raspberry Pi and buffers it.

## Pi Polls Printer over USB

In **nfc.py:**

- **get_credentials()** sends QUERY_CREDENTIALS to the printer over USB

## Printer transfers packet over USB

- If the printer has NFC data buffered at this time, it will be returned to the Pi

## Printer Empties NFC Buffer

- In response to ***erase_credentials()**,* the printer empties the NFC buffer.

## Pi Script Processes Intent…

- **nfc.py** uses *Regular Expressions* to validate and group the message components
- If the 'method' component is confirmed as being a 'pipsta' method, it is used as a lookup in a *Dictionary* to determine the *module* to process the payload. By default, the supported modules are **banner** and **qr**
- The ***send_to_printer()*** method in the target module is invoked. This method performs all the processing that is discussed in ***PIPSTA105*** and ***PIPSTA106***.

## Python Script Sends Print-Job…

- After generating the appropriate image in Pillow, the appropriate Python module sends a sequence of graphical data to the printer.

## Next Steps

The tutorial ***PIPSTA208-Adding New NFC Methods to the Pipsta NFC Server*** will demonstrate how you can extend both the Android app and the Pi host NFC server system to add you own methods.

If you are solely interested in adding new capabilities to the Android app, you may also wish to work through ***PIPSTA010-014*** and the corresponding ***PIPSTA111-114*** tutorials, which culminate in a Secure Printing application.

## 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■**