



# PIPSTA008 – USING NFC WITH PIPSTA

---

**Contents**

Difficulty Level:..... 2

Time to Complete: ..... 2

Who Should Read This Document ..... 2

Introduction ..... 3

Pre-Requisites ..... 3

Getting Started..... 4

Send to Printer ..... 5

    How it Works..... 6

Send to Pi ..... 6

    How it Works..... 7

Adjusting Functionality ..... 8

Next Steps ..... 8

Shutting Pipsta Down Safely ..... 9

## Revision History

Revision	Author	Date	Description
1.0	AH	24/11/14	First Release

## Difficulty Level:



- This tutorial gives a brief introduction to Pipsta's NFC functionality
- Details of the Python script and Android App functionality are covered in a later tutorial.

## Time to Complete:



- This tutorial should take 20-30 minutes.

## Who Should Read This Document

NFC is an important emerging technology and will see increasing use in IoT (Internet of Things) and IoP (Internet of People) applications.

Pipsta features an NFC interface for use with NFC-enabled smart phones, and a basic but versatile Android app has been developed to demonstrate some of these capabilities. This can be downloaded from the Google Play store by visiting:

<https://play.google.com/store/apps/details?id=com.pipsta.pipstanfcprinter>

This document is a useful grounding for those wishing to:

- Understand how NFC technology can deliver IoT and IoP solutions
- Learn how to issue simple print-jobs over NFC
- Demonstrate NFC technology in the classroom
- Implement secure or privileged printing
- Implement their own NFC commands to be processed by their own Python scripts
- Learn how to launch specific functions on the Raspberry Pi using their NFC-enabled smart phone



## Introduction

NFC stands for *Near-Field Communication* and is an intentionally short-range wireless technology. NFC is a *proximity* technology, and has a range of just a few centimetres. This can be contrasted with *vicinity* technologies such as Bluetooth Low Energy (also known as BLE, Bluetooth 4.0 and Bluetooth Smart) which has a range up to several metres.

NFC technology makes a virtue of being short range and therefore less prone to *Man-In-The-Middle* security breaches prevalent in other longer-range wireless technologies. As such, NFC is being widely adopted for payment schemes.

Android, Windows and Blackberry mobile devices have had NFC capability for a considerable amount of time, and –as of iOS8 and iPhone6 – Apple has finally introduced (albeit restricted) NFC capability. At the time of writing, this only provides *Apple Pay* functionality, and –as such—until Apple’s NFC capabilities are unlocked, there can be no iOS counterpart to the Pipsta Android app. Nevertheless, Apple’s introduction of NFC signals that this technology has separate and distinct applicability not delivered by Bluetooth Low Energy beacon technology.

With NFC on the cusp of enjoying cross-platform support, and NFC hardware finding its way into even modestly priced smart phone devices, Pipsta has been designed to take advantage of this contemporary communications interface. Whether you intend to simply familiarise yourself with the provided ready-made applications, or wish to develop your own NFC solutions, this document will provide a firm foundation in this state-of-the-art technology.



**TIP:**

New NFC applications and guides will be added on an ongoing basis, so be sure to check [www.pipsta.co.uk](http://www.pipsta.co.uk) periodically.

## 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)
- **A QR code reading app, such as:**  
<https://play.google.com/store/apps/details?id=me.scan.android.client>
- **An NFC-enabled Android smart-phone with the Pipsta app installed from:**  
<https://play.google.com/store/apps/details?id=com.pipsta.pipstanfcprinter>

For a list of compatible NFC-enabled Android Smart Phones, please refer to:

[http://en.wikipedia.org/wiki/List\\_of\\_NFC-enabled\\_mobile\\_devices](http://en.wikipedia.org/wiki/List_of_NFC-enabled_mobile_devices)



## TIP:

Before committing to buying an NFC enabled Android phone, ensure that the hardware supports the ISO14443 NFC tag type and has an Android OS version that supports the hardware (Android 4.1.2 or later)

○



## TIP:

Pipsta and the Pipsta Android App have been tested with the following devices:

- Samsung Galaxy S3
- Samsung Galaxy S3 Mini
- Samsung Galaxy S4 Mini
- Samsung Galaxy Fame

**At the end of the tutorial, please let us know the details of your Android smart-phone so we can update the above list.**

## 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/nfc**
- 5) From File Manager, press **[F4]** to open LXTerminal and enter the following at the \$ prompt:  
**python pipsta.py start**  
This will start the NFC server daemon and enable NFC communications on the Pipsta.

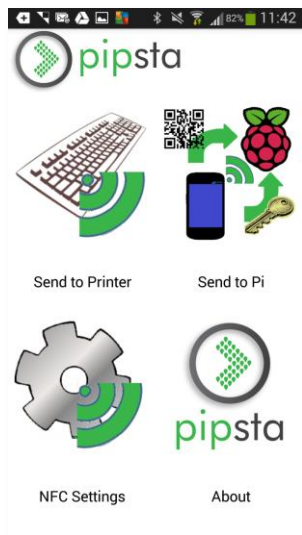




**TIP:**

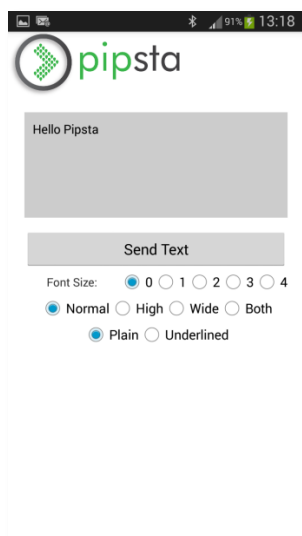
A *daemon* is a program that runs in the background, and so will not appear as a running *application*.

- 6) On the Android phone, go to the *Settings app* and enable NFC
- 7) Now launch the Pipsta App.
- 8) The Pipsta App will present you with a screen as follows:



## Send to Printer

- 1) Click the '*Send to Printer*' button on the app
- 2) Enter some text, in the app text box, followed by 5 carriage returns (↵)





## TIP:

For the time-being, Pipsta NFC communications are limited to 64 bytes, and some of these bytes detail the NFC encoding. Keeping your message <30 characters will ensure that the system does not truncate (shorten) the data printed!

We are working to improve the amount of data that can be sent over NFC .

- 3) Press the 'Back' button on your phone to hide the keyboard:



- 4) Click 'Send Text' on the app
- 5) You will be instructed to 'Tap on Pipsta'. Whilst this message is being displayed, tap the middle of the back of your phone to the green LED on the printer. Your message should be printed on the paper roll.



## TIP:

The distance at which NFC communications will begin to operate is dependent on a number of factors:

- Alignment of coil in smart phone to coil in Pipsta printer
- Rotation of smart phone with respect to Pipsta
- Size of smart phone coil
- Thickness of smart phone case

After a bit of practice, the optimal position will be found.

Note that a 15mm NFC range is not unusual for an optimally aligned smart phone.

Also note that the green LED is not involved in NFC data transfer: it is simply useful as a central target for the phone!

## How it Works

This basic 'Send to Printer' mode demonstrates the ability of the Pipsta to print data sent over NFC *without the involvement of the Raspberry Pi*.



## TIP:

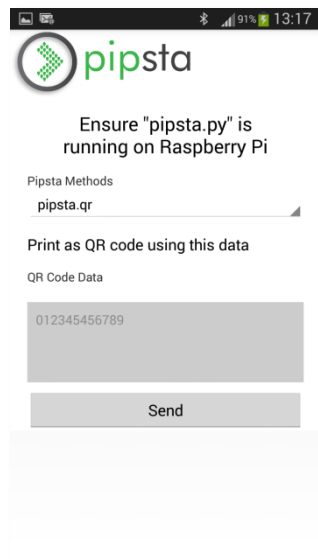
In order to verify that the Raspberry Pi is not involved, try disconnecting the USB cable connecting the Pipsta and the Raspberry Pi from the back of the unit and issuing another *Send to Printer*.


This is useful as a 'sanity-check' to show that NFC and the printer are both working.

## Send to Pi


- 1) From the main screen on the app, click the 'Send to Pi' button
- 2) By default, there are two 'methods' demonstrated by the app. We will first explore the method called 'pipsta.qr'
- 3) Under 'Pipsta Methods', ensure the list box has 'pipsta.qr' selected. If not, click the list and select the correct 'method'





- 4) The screen will also show a text field called QR Code Data. By default, this field will be filled with the following data:  
**012345456789**
- 5) We will use this default data for this example. Press the 'Back' button on your phone to hide the keyboard:  

- 6) Click 'Send', and tap the smart phone to the Pipsta LED as before.
- 7) After a few moments, the LED on the printer should start to flash green.
- 8) A few moments after that, a QR Code should be printed and the LED should stop flashing.
- 9) Press back to exit the Pipsta App and locate and launch the QR Code reader app.
- 10) Holding the QR Code flat, and ensuring your fingers are away from the code itself, use the QR Code app to scan the QR code that was printed.
- 11) Verify that the code matches that sent from the Pipsta App.
- 12) Finally, in LXTerminal on the Raspberry Pi, enter  
**python pipsta.py stop**

to stop the server. Failure to do so may lead to confusion when you start stand-alone applications.

**TIP:**

QR Codes of this size actually require a few millimetres of white space or 'Quiet Zone' around the code, so ensure your fingers are clear of this zone or your QR Code app may not be able to scan the code!

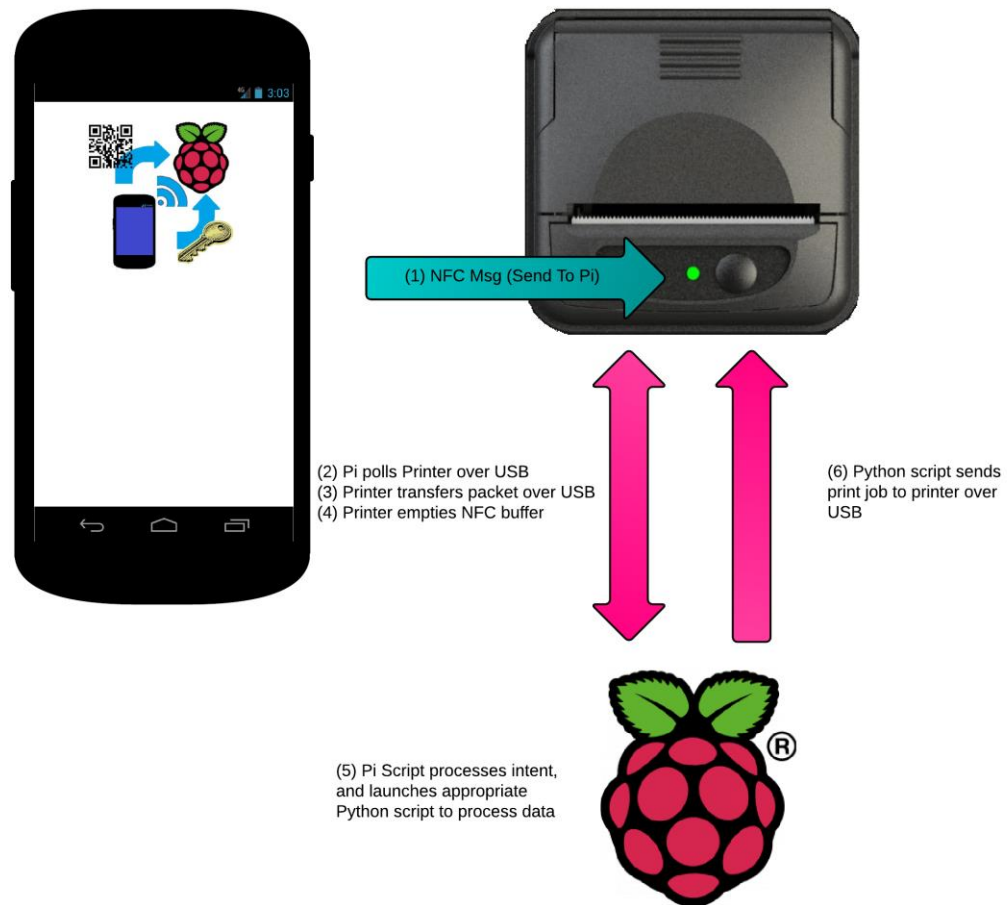
## How it Works

Unlike the 'Send to Printer' mode, this 'Send to Pi' mode *does* involve the Raspberry Pi, and—in fact—forwards the required method and the payload data to the Raspberry Pi **Pipsta.py** script. The script determines which method is required and passes the payload data to the script for processing. In this case, the required method was **Pipsta.qr**, a QR Code generating app similar to that



demonstrated in *PIPSTA006 – Pipsta QR Codes*, but this could equally have been a banner print, or any other bespoke method defined by the user.

The system diagram is as follows:



## Adjusting Functionality

The Pipsta app has designed to be *extensible*, allowing users to add their own bespoke methods to the application. See *Next Steps* below to discover how to extend the app's functionality to include your own methods to be handled by your own Python scripts.

## Next Steps

If this is the end of your session, see the section entitled *Shutting Pipsta Down Safely*, or –to continue on to learn how the Script works with the following tutorial:

- **PIPSTA108 –NFC Python Code Tutorial**

Thereafter, you can progress to:

- **PIPSTA208 – Adding New NFC Methods to the Pipsta Server**





## 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■