How to setup the Raspberry Pi for the FS Token System

Installation

Downloaded and installed Raspbian Wheezy release 2013-02-09, it's all documented on the download page.

At first, Raspberry Pi would not boot, therefore I modified /boot/config.txt

```# I had uncommented this line, but afterwards I commented it again because it worked fine without
#hdmi_safe=1
# This line might not really be needed
hdmi_mode=16```

Plugged the memory card into the Raspberry Pi, started up and modified following settings in raspi-config:

| expand-rootfs |   |
| configure_keyboard | US layout |
| change_locale | en_US |
| change_timezone | Kolkata |
| SSH | Enabled |
| Password of pi user | Changed and documented in KeePassX |
| Boot Behaviour | Straight to desktop |

Setup apt-cacher, /etc/apt/apt.conf:

```Acquire::http { Proxy "http://192.168.28.2:3142/" };```

Then installed a few packages

```# apt-get update
# apt-get install git
# apt-get install apache2 php5 libapache2-mod-php5 mysql-server-5.5 phpmyadmin
# apt-get install matchbox
# apt-get install x11-xserver-utils
# apt-get install unclutter```

I chose a MySQL root password and documented in KeePassX

Configured PHPMyAdmin with Apache2, when requested

Installation of the Token System software

```$ ssh-keygen -t dsa```

In .ssh/config
host bluelight
hostname bluelightav.org
port 2222
user git

Obtain your public key from `.ssh/id_dsa.pub` and add it to the git repository

$ git clone bluelight:tokensystem
$ sudo ln -s /home/pi/tokensystem/ /var/www/

In PHPMyAdmin, created new user:

username: fstokensystem
password: documented in KeePassX
selected option "Create database with same name and grant all privileges"

Imported 'fstokensystem.sql' (attached to this wiki page) into the new 'fstokensystem' db.

Edited `~/tokensystem/vars.php`

$db_pass= // set to fstokensystem MySQL user password

Installation of the GPIO software

Install the wiringPi library:

$ git clone git://git.drogon.net/wiringPi
$ cd wiringPi/
$ git pull origin
$ ./build

The BASH script that makes use of the above library, and polls for the button press is included in the 'tokensystem' Git repository.

Optional LXDE Configuration

This part is not really necessary as we are going to disable LXDE desktop environment, but I have done it and it might be nice to have as a backup.

Opened Midori and set the following as homepage: http://localhost/tokensystem/display.php

$ mkdir .config/autostart

Edit `.config/autostart/tokensystem.desktop`

[Desktop Entry]
Name=Token System Display
Comment=Fullscreen Display of the Token System
Exec=/usr/bin/midori -a http://localhost/tokensystem/display.php -e Fullscreen
Terminal=false
MultipleArgs=true
Type=Application
Categories=Application;Other;
StartupNotify=true
MimeType=x-scheme-handler/indiecity;

Edit /etc/lightdm/lightdm.conf, and add in 'SeatDefaults'
Edit .config/autostart/gpio-button.desktop

[Desktop Entry]
Name=Token System GPIO button
Comment=GPIO reader BASH script for dispenser button
Exec=/home/pi/tokensystem/gpio_read/gpio_read.sh
Terminal=false
MultipleArgs=true
Type=Application
Categories=Application;Other;
StartupNotify=true
MimeType=x-scheme-handler/indiecity;

X Startup configuration

Create file called ‘wm_midori’ in the home folder

#!/bin/sh
sudo xset -dpms
sudo xset s off
matchbox-window-manager &
/home/pi/tokensystem/gpio_read/gpio_read.sh &
unclutter &
while true; do
    midori -e Fullscreen -a http://localhost/tokensystem/display.php
done

then

$ chmod +x wm_midori

Edit /etc/inittab

#1:2345:respawn:/sbin/getty --noclear 38400 tty1 # comment out this line
1:2345:respawn:/bin/login -f pi tty1 >/dev/tty1 >/dev/tty1 2>&1 # and add this one instead

Edit .profile (in home folder), and add at the end of the file

xinit ./wm_midori

Run raspi-config once again and disable desktop on boot

Shutdown Configuration

Run visudo, and after

root ALL=(ALL) ALL

add the following
# Allow PHP script to run poweroff
www-data ALL= NOPASSWD:/sbin/poweroff

## Disable Screen Blanking

This part might not be necessary as the wm_midori script is supposed to already take care of it, but if it does not, try the following:

**Edit /etc/kbd/config**

```bash
BLANK_DPMS=off
POWERDOWN_TIME=0
BLANK_TIME=0
```

**Edit /etc/matchbox/kbdconfig**

```
BLANK_TIME=0
POWERDOWN_TIME=0
BLANK_DPMS=off
```

and

```bash
# chmod -x /usr/bin/xdg-screensaver
```

## Network Configuration

**Edit /etc/network/interfaces**

```bash
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.0.244
netmask 255.255.255.0
gateway 192.168.0.1

# The below is not really needed, but it was already there
allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

**Edit /etc/resolv.conf**

```bash
nameserver 208.67.222.222
```

## How to add clients

### Desk Client

To add a desk client, create a launcher on the panel or Desktop to the following command:
wget http://192.168.0.244/tokensystem/call.php?counter=NUMBER&auth=MD5HASH -O -

Replace NUMBER with the desk number and MD5HASH with the hash of the authentication password, you can find the HASH in KeePassX.

**Dispenser Client**

To add a token dispenser client, create a launcher on the panel or Desktop to the following command:

```
wget "http://localhost/tokensystem/call.php?dispenser=NUMBER&auth=MD5HASH" -O -
```

Replace NUMBER with the dispenser number and MD5HASH with the hash of the authentication password, you can find the HASH in KeePassX.

**Shutdown command**

To add a shutdown command on a machine, create a launcher on the panel or Desktop to the following command:

```
wget http://192.168.0.244/tokensystem/shutdown.php?auth=MD5HASH -O -
```

Replace MD5HASH with the hash of the authentication password, you can find the HASH in KeePassX.