Raspberry Pi en mode kiosque

Objectif

Afficher une page web en plein écran grâce à un Raspberry Pi.

Prérequis

Un Raspberry Pi avec une carte SD de 2 Go minimum, son alimentation, un câble réseau RJ45, un clavier et une souris (optionnels). Un moniteur ou un téléviseur avec une entrée HDMI (ou un adaptateur HDMI/DVI ou VGA).

Nous considérerons que l'utilisateur par défaut est **pi** et que l'adresse réseau est attribuée par DHCP.

Choisissez votre éditeur de texte favori.

Installation

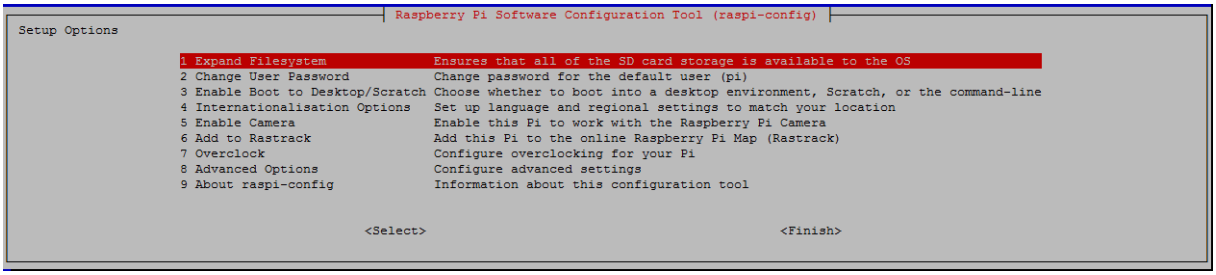
Télécharger la distribution Raspbian[1]

Si vous êtes sous Windows, utilisez Win32 Disk Imager[2]

Sous Linux utilisez la commande dd
dd bs=4M if=nom_image_distribution.img of=/dev/sdx (avec sdx selon votre carte SD)

Configuration de base

Lors du 1er démarrage, le script **raspi-config** s'exécute automatiquement pour paramétrer un certain nombre de choses, voici le minimum à faire :



1 – Expand Filesystem : Agrandir les partitions pour utiliser l'intégralité de l'espace de la carte SD.
2 – Change User Password : Modifier le mot de passe de l'utilisateur **pi**
4 - Internationalisation Options :

- Change Locale : **fr_FR@UTF8**
- Change Timezone : **Europe/Paris**
- Change Keyboard Layout : **fr**

8 - Advanced Options :

- Hostname : **dashboard**
- Memory Split : **128**
- SSH : **Enable**

Les valeurs ci-dessus sont données à titre indicatif, adaptez selon votre convenance.
Redémarrez pour appliquer les modifications sur le système de fichiers.

Paramétrage

Personnellement, j'aime bien travailler sur un système à jour donc je vous conseille de faire :

sudo rpi-update

Puis un reboot pour prendre en compte la mise à jour du firmware.

sudo apt-get update
sudo apt-get –y upgrade
sudo apt-get –y clean

Afin de gagner de l'espace nous allons supprimer quelques paquets et fichiers inutiles :

apt-get remove wolfram-engine

Supprimez le répertoire **python_games** dans **/home/pi/**

Nous allons installer quelques paquets essentiels

apt-get install x11-xserver-utils fbi xscreensaver

Pour la suites des opérations, je me suis basé sur une documentation trouvée sur Internet, merci à son auteur[3]

Ajoutez les lignes suivantes dans le fichier **/home/pi/.xinitrc**

xset s off
xset -dpms
xset s noblank

exec /etc/alternatives/x-session-manager

Modifiez le fichier **/etc/kbd/config**

..
BLANK_TIME=0
...
BLANK_DPMS=off
...
POWERDOWN_TIME=0
...

Connectez-vous en mode graphique avec **startx** pour désactiver l'économiseur d'écran.

Nous allons mettre en place un « splashcreen » rudimentaire avec **fbi**[4]

Copiez une image de démarrage en format PNG nommée **splash.png** dans le répertoire **/etc**
Créez un fichier **asplashscreen** dans le répertoire **/etc/init.d** contenant :

```
#! /bin/sh
### BEGIN INIT INFO
# Provides:         asplashscreen
# Required-Start:
# Required-Stop:
# Should-Start:
# Default-Start:     S
# Default-Stop:
# Short-Description: Show custom splashscreen
# Description:       Show custom splashscreen
### END INIT INFO

do_start () {

   /usr/bin/fbi -T 1 -noverbose -a /etc/splash.png
   exit 0
}

case "$1" in
  start|"")
    do_start
    ;;
  restart|reload|force-reload)
    echo "Error: argument '$1' not supported" >&2
    exit 3
    ;;
  stop)
    # No-op
    ;;
  status)
    exit 0
    ;;
  *)
    echo "Usage: asplashscreen [start|stop]" >&2
    exit 3
    ;;
esac

:
```

Rendre le fichier exécutable et l'installer comme service

```
sudo chmod a+x /etc/init.d/asplashscreen
sudo insserv /etc/init.d/asplashscreen
```

Afin de lancer le navigateur automatiquement nous allons modifier le fichier **/etc/xdg/lxsession/LXDE/autostart** de la façon suivante :

```
#@lxpanel --profile LXDE
#@pcmanfm --desktop --profile LXDE
#@xscreensaver -no-splash
@midori -e Fullscreen -a http://www.mapage.com
```

Midori [5] est un navigateur léger basé sur Webkit qui supporte HTML5, CSS3, il est installé par défaut sur Raspbian et bien entendu open-source. ;-)

Pour activer l'autologin et réduire le nombre de consoles à deux , nous allons modifier le fichier **/etc/inittab**

Remplacer

1:2345:respawn:/sbin/getty --noclear 38400 tty1

par

1:2345:respawn:/sbin/getty --autologin pi --noclear 38400 tty1

et ajouter # devant les lignes

#3:23:respawn:/sbin/getty 38400 tty3
#4:23:respawn:/sbin/getty 38400 tty4
#5:23:respawn:/sbin/getty 38400 tty5
#6:23:respawn:/sbin/getty 38400 tty6

Ajoutez **startx -- -nocursor** à la fin du fichier **/home/pi/.bashrc** pour lancer X sans le pointeur de la souris.

Pour la suite des évènements nous allons devoir attribuer un mot de passe à root car l'autologin lancera automatiquement le navigateur sous la session de l'utilisateur pi et ça peut être problématique pour effectuer des modifications à distance via SSH.

sudo passwd root

Pour éviter l'affichage du curseur de la console sous X, il faut ajouter **vt.global_cursor_default=0** à la fin du fichier **/boot/cmdline.txt**.

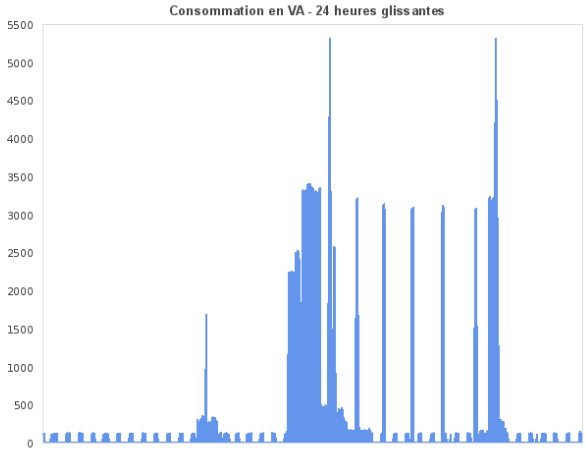
Pour éliminer les bandes noires sur les côtés, notamment lors d'une connexion sur un téléviseur via un câble HDMI, ajustez les paramètres suivants :

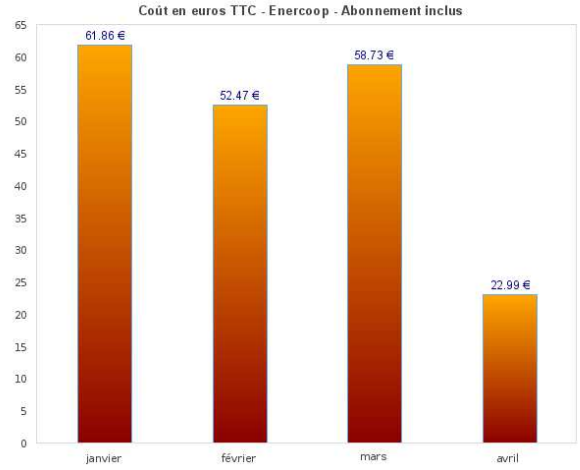
hdmi_mode=16
hdmi_drive=2
disable_overscan=1

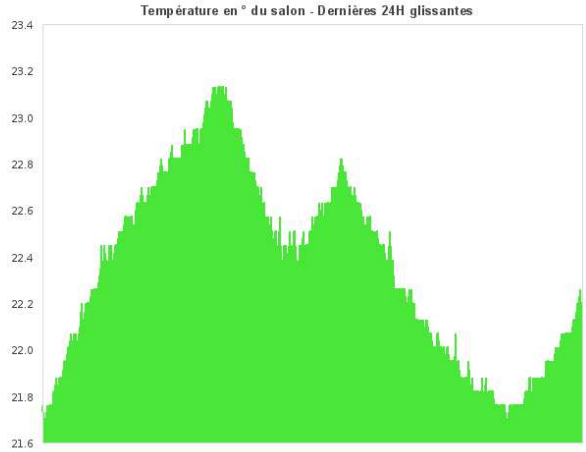
Redémarrez le Raspberry Pi et profitez de votre réalisation…

Résultat

Voici un exemple d'affichage que j'ai mis en place pour le suivi de ma consommation électrique.







Conclusion

Grâce au Raspberry Pi et Linux, vous êtes en mesure de mettre en place un système d'affichage plein écran administrable à distance via SSH pour une somme modique.

Références

(1) http://www.raspberrypi.org/downloads/
(2) http://sourceforge.net/projects/win32diskimager/
(3) http://www.oceandatarat.org/?p=702
(4) http://www.edv-huber.com/index.php/problemloesungen/15-custom-sp
(5) http://www.midori-browser.org/