

11/16/24 Version *kiosk\_flip28.py*

There is no change in the python code. The change is in the autoexec.sh file that launches the program on power-up.

This version adds the middle mouse button, mouse button 2, which will flip the screen just as the doorbell button does.

Added the follow lines of code to version 27:

```
if event.button == 2:  
    print ('middle mouse button')  
if disp_mode ==english:  
    disp_mode = spanish  
else:  
    disp_mode = english
```

1/4/24

The latest version is kiosk\_flip27.py

In this version, the English display is set by a left mouse button and the Spanish display is set by the right mouse button.

9/14/23

The program will flip between displaying Spanish and English .png files when either any mouse button is pressed or the “door bell” momentary push button switch is pressed.



It runs on a Raspberry Pi and uses pygame. I have run it in Windows after pygame has been installed.

Files needed in the */home/pi/kiosk* folder:

*kiosk\_flip.py*

*ResouceEnglish.png*

*ResourceSpanish.png*

The artwork of the two Resource files is 960x720

On the Raspberry Pi, set the display to 1280x1024

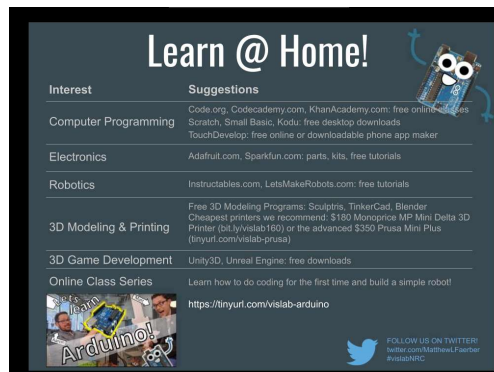
sudo raspi-config

7 Advanced Options

A5 Resolution


DMT Mode 35 1280x1024 60Hz 5:4


If the resolution is changed to **Mode 16 1024 x 768**, the display is closer to full screen:



**Learn @ Home!**

Interest	Suggestions
Computer Programming	Code.org, Codecademy.com, KhanAcademy.com: free online classes Scratch, Small Basic, Kodu: free desktop downloads TouchDevelop: free online or downloadable phone app maker
Electronics	Adafruit.com, Sparkfun.com: parts, kits, free tutorials
Robotics	Instructables.com, LetsMakeRobots.com: free tutorials
3D Modeling & Printing	Free 3D Modeling Programs: Sculpt3D, TinkerCad, Blender Cheapest printers we recommend: \$180 Monoprice MP Mini Delta 3D Printer ( <a href="http://bit.ly/vislab160">bit.ly/vislab160</a> ) or the advanced \$350 Prusa Mini Plus ( <a href="http://tinyurl.com/vislab-prusa">tinyurl.com/vislab-prusa</a> )
3D Game Development	Unity3D, Unreal Engine: free downloads
Online Class Series	Learn how to do coding for the first time and build a simple robot <a href="https://tinyurl.com/vislab-arduino">https://tinyurl.com/vislab-arduino</a>

 **Arduino!**

 FOLLOW US ON TWITTER:  
[@matthewj\\_foster](https://twitter.com/MatthewJ_Foster)  
[@vislabNCC](https://twitter.com/vislabNCC)

The switch is connected to pins 12 and 14 on the Rasp Pi as shown above.

To run the program, switch to the */home/pi/kiosk* folder.

Enter: ***sudo python kiosk\_flipx26.py***.

**Note:** we need to use ***sudo*** because the ***RPi.GPIO*** library was imported.

When running the FULLSCREEN version, you can do nothing else. To get out of the FULLSCREEN mode, you have to reboot the Raspberry Pi.

Rolf Gieger developed a Linux script to launch to program that allows the user to either:

- 1-Do nothing and have the system run in full screen mode.
- 2-Allow the user to select to run the non-full screen mode so the system can be modified.

This script is the ***autoexec.sh*** file

To set up the program to run when the Raspberry Pi is powered up. Follow the procedure in the link below:

[https://dunn-itvideoservicesllc.yolasite.com/resources/RaspberryPi/Run\\_Program\\_on\\_startup.pdf](https://dunn-itvideoservicesllc.yolasite.com/resources/RaspberryPi/Run_Program_on_startup.pdf)

Briefly:

- 1- In the */home/pi/.config* folder create a new folder named ***autostart***.
- 2- In the newly created ***autostart*** folder, create a file named ***.desktop*** with the following 3 lines:

```
[Desktop Entry  
Type = Application  
Exec = lxterminal -e ./autoexec.sh
```

The ***autoexec.sh*** file must be in the */home/pi* folder with the two lines shown below and must be executable. (chmod +x)

---

```
#!/bin/bash
```

```
cd kiosk
```

```
function choose_script() {  
  while true; do  
    echo "Select a script to run:"  
    echo "1. kiosk_flip26.py"  
    echo "2. kiosk_flip26x.py"  
    read -r -t 5 -p "Enter your choice (1 or 2): " choice  
  
    # Check if user input is empty or not a number  
    if [[ -z "$choice" || ! "$choice" =~ ^[1-2]$ ]]; then  
      if [[ -z "$choice" ]]; then  
        echo "No input received. Defaulting to 1."  
        choice=1  
      fi  
    fi  
  done  
}
```

```
break
else
  echo "Invalid input. Please enter 1 or 2."
  continue
fi
fi

break
done

case $choice in
  1)
    echo "Running kiosk_flip26.py..."
    sudo python kiosk_flip.py
    ;;
  2)
    echo "Running kiosk_flip26x.py..."
    sudo python kiosk_flip26x.py
    ;;
esac
}

choose_script
```

---

Any changes should be developed and tested using the x version of the file. Ie ***kiosk\_flipx26.py***. See below.

The ***kiosk\_flip.py*** file is the same as the ***kiosk\_flip26.py***. The files with x in them are not the FULLSCREEN versions.

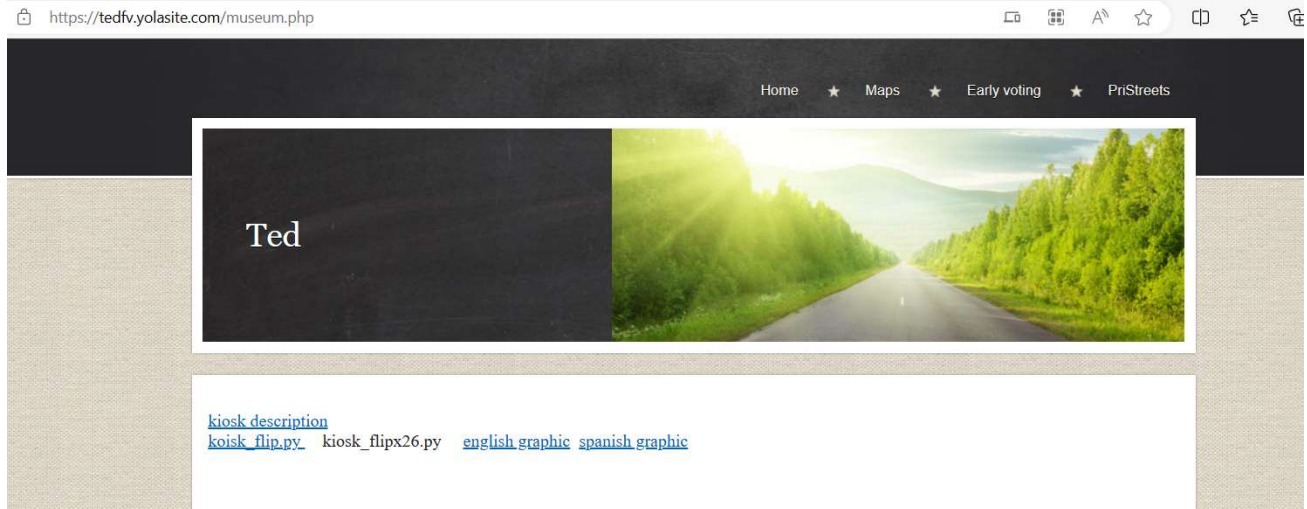
For example the file ***kiosk\_flipx26.py*** is identical to the ***kiosk\_flip26.py*** file except that the line:

***DISPLAYSURF = pygame.display.set\_mode((960, 720), pygame.FULLSCREEN)***

Is commented

***#DISPLAYSURF = pygame.display.set\_mode((960, 720), pygame.FULLSCREEN)***

The files can be downloaded from: <https://tedfv.yolasite.com/museum.php>



To download the graphics file, click on the link. When it is displayed in the browser right click on the image to download it.

## Appendix 1 Listing of the *kiosk\_flip.py* file.

### Notes:

*kiosk\_flip.py* is the same as *kiosk\_flip28.py*. These files are the full screen versions.

*kiosk\_flipx.py* is the same as *kiosk\_flip28x.py*. These files are the non-full screen versions.

```
## Version 27 fixes the english screen to the left mouse  
button  
## and the spanish screen to the right button  
## Version 28 adds mousebutton 3 to flip the screen
```

```
from sys import argv      #used to get file name  
import os  
from time import *  
import time  
import threading  
from datetime import datetime  
  
import RPi.GPIO as GPIO  
GPIO.setmode(GPIO.BCM)  
  
GPIO.setup(18,GPIO.IN, pull_up_down=GPIO.PUD_UP)  
  
ver = '28'  
  
script = argv      # This is the file name  
  
print ('Version: '+ver)
```

```
#####
```

```
#####  
#
```

```

#Define Colours
WHITE = (255,255,255)
BLUE = (0,0,255)
BLACK = (0,0,0)
GRAY = (128, 128, 128)
MAROON = (128, 0, 0)
NAVYBLUE = (0, 0, 128)
OLIVE = (128, 128, 0)
PURPLE = (128, 0, 128)
TEAL = (0,128,128)
PINK = (226,132,164)
MUTEDBLUE = (155,182,203)
PLUM = (221,160,221)
#####
#

#####

import pygame
pygame.init()

screen = pygame.display.set_mode((0,800)) # Sets the
screen size
screen.fill((0,0,0)) #Fills the bbackground with black

pygame.display.set_caption(str(script)+' '+ ver) # Title
of window
spanish =
pygame.image.load('ResourceSpanish.png').convert()
english =
pygame.image.load('ResourceEnglish.png').convert()
pygame.display.set_mode((960,720),0,32) ## sets full screen
DISPLAYSURF = pygame.display.set_mode((960, 720),
pygame.FULLSCREEN)
pygame.mouse.set_visible(False)

def show_display(message):

    font = pygame.font.Font(None,30)
    text = font.render(message, 1,OLIVE)
    screen.blit(text,(900,850)) ### Hor.Vert

```

```
disp_mode= spanish
```

```
ti = 0
```

```
if disp_mode == spanish:  
    message = 'Click on the mouse for English'  
else:  
    message = 'Click en el mouse para espanol'
```

```
#####
```

```
try:  
    while True:  
  
        screen.fill(BLACK)  
        screen.blit(disp_mode,(10,10))  
        if disp_mode == spanish:  
            message = 'Click on the mouse for English'  
        else:  
            message = 'Click on the mouse for Spanish'  
  
        show_display(message)  
  
        pygame.display.update()  
        time.sleep(ti)  
        for event in pygame.event.get():  
            if event.type == pygame.MOUSEBUTTONDOWN:  
                if event.button == 1:  
                    print ('left mouse button')  
                    disp_mode = english  
                if event.button == 3:  
                    print ('right mouse button')  
                    disp_mode = spanish  
  
            if event.button == 2:  
                print ('middle mouse button')
```

```

        if disp_mode ==english:
            disp_mode = spanish
        else:
            disp_mode = english

    print("MOUSEBUTTONDOWN event occured")
    print('disp_mode = ',disp_mode)
    show_display(message)

#####
    if GPIO.input(18) == 0:
        print ('Button Pressed')

        if disp_mode ==english:
            disp_mode = spanish
        else:
            disp_mode = english

        show_display(message)

        time.sleep(.1)

#####

    #screen.fill(BLACK)
    #screen.blit(english,(500,90))
    #pygame.display.update()

    time.sleep(ti)

except (KeyboardInterrupt, SystemExit):
    print('The program was manually stopped')
    quit()

'''
    mouse_presses = pygame.mouse.get_pressed()
    if mouse_presses[0]:
        print("Left Mouse Key is being pressed")
        if '''

```