**Teach-In 2014 with Raspberry Pi: Part 4**

by Mike and Richard Tooley

These are text files of the source code listings printed in EPE.

They appear in the same order as in the articles.

Separate listings are split by four empty lines.

import tkinter

import tkinter

root = tkinter.Tk()

w = tkinter.Label(root, text="Here is some text")

import tkinter as tk

import tkinter as tk

root = tk.Tk()

w = tk.Label(root, text="Here is some more text")

from tkinter import \*

from tkinter import \*

root = Tk()

w = Label(root, text="Hello, world!")

w.pack()

root.mainloop()

from tkinter import \*

root = Tk()

w = Label(root, text="Teach-In with Raspberry Pi")

w.pack()

root.mainloop()

from tkinter import \*

import time

title = "Time Box".title()

msg = Message(text= "%s" % time.asctime())

msg.config(bg='yellow', font=('times', 16, 'bold'))

msg.pack()

mainloop()

from tkinter import \*

import time

title = "Time Box".title()

msg = Message(text= "%s" % time.asctime())

msg.config(bg='yellow', font=('times', 16, 'bold'))

msg.pack()

mainloop

import time

import os

import RPi.GPIO as GPIO

from gpiospiadc import \*

GPIO.setmode(GPIO.BOARD)

DEBUG = 1

# Define pins on the GPIO connector

SPICLK = 23

SPIMISO = 21

SPIMOSI = 19

SPICS = 24

# Set up the SPI

GPIO.setup(SPIMOSI, GPIO.OUT)

GPIO.setup(SPIMISO, GPIO.IN)

GPIO.setup(SPICLK, GPIO.OUT)

GPIO.setup(SPICS, GPIO.OUT)

sensor\_adc = 0

while True:

sensor\_value = readadc(sensor\_adc, SPICLK, SPIMOSI, SPIMISO, SPICS)

sensor\_voltage = (sensor\_value \* 3.3)/1024

print("Analogue input voltage = %5.3f V" % (sensor\_voltage))

time.sleep(0.5)

#!/usr/bin/env python

import time

import os

import RPi.GPIO as GPIO

from gpiospiadc import \*

GPIO.setmode(GPIO.BOARD)

DEBUG = 1

# Define pins on the GPIO connector

SPICLK = 23

SPIMISO = 21

SPIMOSI = 19

SPICS = 24

# Set up the SPI

GPIO.setup(SPIMOSI, GPIO.OUT)

GPIO.setup(SPIMISO, GPIO.IN)

GPIO.setup(SPICLK, GPIO.OUT)

GPIO.setup(SPICS, GPIO.OUT)

sensor\_adc = 0

while True:

sensor\_value = readadc(sensor\_adc, SPICLK, SPIMOSI, SPIMISO, SPICS)

sensor\_voltage = (sensor\_value \* 3.3)/1024

print("Analogue input voltage = %5.3f V" % (sensor\_voltage))

time.sleep(0.5)

#!/usr/bin/env python

from tkinter import \*

import time

import os

import RPi.GPIO as GPIO

from gpiospiadc import \*

root = Tk()

root.title("VOLTS")

voltage1 = ''

meter = Label(root, font=('arial', 36, 'bold'), bg='green')

meter.pack(fill=BOTH, expand=1)

GPIO.setmode(GPIO.BOARD)

DEBUG = 1

# Define pins on the GPIO connector

SPICLK = 23

SPIMISO = 21

SPIMOSI = 19

SPICS = 24

# Set up the SPI

GPIO.setup(SPIMOSI, GPIO.OUT)

GPIO.setup(SPIMISO, GPIO.IN)

GPIO.setup(SPICLK, GPIO.OUT)

GPIO.setup(SPICS, GPIO.OUT)

sensor\_adc = 0

def measure():

global voltage1

# get the current voltage from the SPI device

sensor\_value = readadc(sensor\_adc, SPICLK, SPIMOSI, SPIMISO, SPICS)

sensor\_voltage = (sensor\_value \* 3.3)/1024

voltage2 = sensor\_voltage

sv = '%.3f' % sensor\_voltage

# if the voltage has changed, update it

if voltage2 != voltage1:

voltage1 = voltage2

meter.config(text=sv)

# update the display every 200ms

meter.after(200, measure)

measure()

root.mainloop()

sudo apt-get update

sudo apt-get install x11vnc

[Desktop Entry]

Encoding=UTF-8

Type=Application

Name=X11VNC

Exec=x11vnc –forever –passwd epe2013 –display :0 –ultrafilexfer

StartupNotify=false

Hidden=false

sudo raspi-config

sudo reboot

192.168.1.7:0