Teach-In 2014 with Raspberry Pi: Part 8by Mike and Richard TooleyThese 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.# Stepper motor test routine for Python 3# One quarter revolution CW in 10s followed # by one quarter revolution ACW in 10s# Motor left in standby mode# Import the required librariesimport RPi.GPIO as GPIOimport time# Define pin numbering conventionGPIO.setmode(GPIO.BOARD)# Set up pins 11, 12 and 13 as outputsGPIO.setup(11,GPIO.OUT) # STEPGPIO.setup(12,GPIO.OUT) # DIRECTIONGPIO.setup(13,GPIO.OUT) # INH# Take INH high to energise the motorGPIO.output(13,1)# Get ready to startprint('Motor starting in 4 seconds ...')time.sleep(4)# Quarter revolution ACWGPIO.output(12,0) # Direction = CWn = 0while n < 50: GPIO.output(11,1) time.sleep(0.1) GPIO.output(11,0) time.sleep(0.1) n = n + 1# Quarter revolution CWGPIO.output(12,1) # Direction = ACWn = 0while n < 50: GPIO.output(11,1) time.sleep(0.1) GPIO.output(11,0) time.sleep(0.1) n = n + 1# Take INH low to release the motor# and place the NJM2671 in standyGPIO.output(13,0)# Finally clean up the GPIOGPIO.cleanup()# Stepper motor application for Python 3# Control via radio button dialogue# Uses full step mode (no link in place)# Import the required librariesimport RPi.GPIO as GPIOimport timefrom tkinter import \*# Define the pin numbering conventionGPIO.setmode(GPIO.BOARD)# Set up pins 11, 12 and 13 as outputsGPIO.setup(11,GPIO.OUT) # STEPGPIO.setup(12,GPIO.OUT) # DIRECTIONGPIO.setup(13,GPIO.OUT) # INHdef stop(): # Release the motor to standby GPIO.output(13,0) time.sleep(1)def cw(): # Full revolution ACW # Take INH high to energise the motor GPIO.output(13,1) time.sleep(1) GPIO.output(12,0) # Direction = CW n = 0 while n < 200: GPIO.output(11,1) time.sleep(0.01) GPIO.output(11,0) time.sleep(0.01) n = n + 1def acw(): # Full revolution CW # Take INH high to energise the motor GPIO.output(13,1) time.sleep(1) GPIO.output(12,1) # Direction = ACW n = 0 while n < 200: GPIO.output(11,1) time.sleep(0.01) GPIO.output(11,0) time.sleep(0.01) n = n + 1def sel(): option = str(var.get()) if option == '1': stop() if option == '2': cw() if option == '3': acw()# Dialogue window with radio buttonsroot = Tk()root.title('Motor')root.minsize(140,100)var = IntVar()# First radio buttonR1 = Radiobutton(root, text='Release', fg='red', variable=var, value=1)R1.pack(anchor=W)# Second radio buttonR2 = Radiobutton(root, text='Clockwise', fg='blue', variable=var, value=2)R2.pack(anchor=W)# Third radio buttonR3 = Radiobutton(root, text='Anticlockwise', fg='blue', variable=var, value=3)R3.pack(anchor=W)# Go buttonG = Button(root, text='GO', fg='green', command=sel)G.pack()root.mainloop()GPIO.cleanup()sudo apt-get updatesudo apt-get dist-upgradesudo raspi-config raspistill -o myphoto.jpgraspistill -o mphoto.jpg Ðt 5000raspivid -o myvideo.h264 -t 10000raspivid -t 180000 -d raspistill -o timelapse%04d.jpg -t 3600000 -tl 10000sudo raspistill -n -o /var/www/webcam.jpg -w 320 -h 240 -q 50 -t 0 -tl 5000nohup sudo raspistill -n -o /var/www/webcam.jpg -w 320 -h 240 -q 50 -t 0 -tl 5000<html><head><script type="text/javascript"> <!-- function WebcamRefresh() { var timestamp = new Date().getTime(); document.getElementById("webcam").src = "webcam.jpg?" + timestamp; setTimeout("WebcamRefresh();", 5000); } onload = WebcamRefresh; --></script></head><body> <p><h1>Pi Webcam</h1><hr/></p> <p><h3>Welcome to my live Pi webcam page!</h3></p> <p>The webcam image will automatically refresh every 5 seconds...</p> <p><img id="webcam"src="webcam.jpg" ></img></p> </body></html>2