INTERNET OF THINGS – IT134IU

LAB 4 and Lab 4 Extra

Instructions: Please follow the steps:
1 – Students work in groups. Please answer the questions clearly. Remember to include your name and your student ID.
2 - Each group leader (ONLY the group leader, please!) submits the report in Pdf format before the deadline.

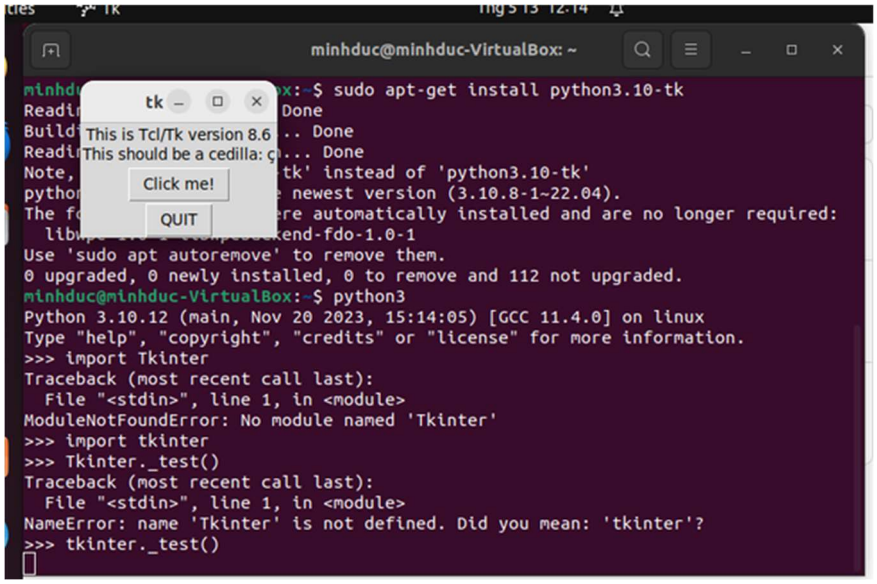
Group Name:			
No	**ID**	**Name**	**No Contribution (0 point)**
1	ITITIU21045	Nguyễn Minh Đức	16%
2	ITITIU21123	Trần Hoàng Thế Bảo	16%
3	ITITIU21217	Đỗ Đức Huy	16%
4	ITITIU21076	Bùi Đức Mạnh	16%
5	ITITIU21347	Phạm Quang Vinh	16%
6	ITITIU20021	Hàng Huỳnh Công Thuận	20%

LAB 4 and LAB 4 EXTRA

1. **Activity 2**
   a) Take the result screenshot photos.
   b) Answer all the questions

```
import pyfirmata  # import lib of py#rmata
import time as wait     # import lib of py#rmata
board = pyfirmata.Arduino('/dev/ttyACM0')# de#ne COM port of Arduino
red_pin = board.get_pin('d:7:o')# assign digital pin 7 as an output
green_pin = board.get_pin('d:6:o')# assign digital pin 6 as an output
while True:       # in#nite loop
    red_pin.write(1)# write ‘1’ on pin 7
    green_pin.write(1)# write ‘1’ on pin 6
    wait.sleep(0.5)# delay of 0.5 Sec
    red_pin.write(0)#write ‘0’ on pin 7
    green_pin.write(0)# write ‘0’ on pin 6
    wait.sleep(0.5)# delay of 0.5 Sec
```

2. **Activity 3**
   a) Take the result screenshot photos



   b) Answer all the questions

```
import tkinter
import pyfirmata
import time as wait
# Associate port and board with py"rmata
board = pyfirmata.Arduino('/dev/ttyACM0')
wait.sleep(5) # delay of 5Sec
```

```
    led_Pin = board.get_pin('d:7:o') # connect led to pin 7 and used as output
    def call_LED_BLINK():
        button.config(state = tkinter.DISABLED)
        led_Pin.write(1) # make led_Pin to HIGH
        print('LED at pin7 is ON') # print on terminal
        wait.sleep(5) # delay of 5 sec
        print('LED at pin 7 is OFF') # print on terminal
        led_Pin.write(0) # make led_Pin to LOW
        button.config(state=tkinter.ACTIVE)
    # Initialize main windows with title and size
    TOP = tkinter.Tk()
    TOP.title("Blink LED using button")
    TOP.minsize(300,30)
    # Create a button on main window and associate it with above method
    button = tkinter.Button(TOP, text="Press start to blink", command =
    call_LED_BLINK)
    button.pack()
    TOP.mainloop()
```

Control led brightness:

```
import tkinter # add Tkinter library
import pyfirmata # add py"rmata library
import time as wait # add time library
board = pyfirmata.Arduino('/dev/ttyACM0')
wait.sleep(5) # delay of 5 Sec
led_Pin = board.get_pin('d:6:p') # connect led to pin 7 and used as output
def call_led_blink_pwm():
        time_Period = time_Period_Entry.get()
        time_Period = float(time_Period)  led_Brightness =
brightness_Scale.get()
        led_Brightness = float(led_Brightness / 100.0)
        print(led_Brightness)
        button.config(state=tkinter.DISABLED)
        led_Pin.write(led_Brightness)
        print ('LED brightness control') # print on terminal
        wait.sleep(time_Period)
        led_Pin.write(0) # make led_Pin to LOW
        button.config(state=tkinter.ACTIVE)
```

TOP = tkinter.Tk()
time_Period_Entry = tkinter.Entry(TOP, bd=7, width = 30)
time_Period_Entry.pack()time_Period_Entry.focus_set()
brightness_Scale = tkinter.Scale(TOP, from_=0, to=100,
orient=tkinter.VERTICAL)
brightness_Scale.pack()
button = tkinter.Button(TOP, text="Start", command = call_led_blink_pwm)
button.pack()
TOP.mainloop()

Link to video of activity 2 and activity 3:

https://drive.google.com/drive/folders/1aPQQJVIRZLKLE5PBTYC6S2q7MOnnwqhn?usp=sharing

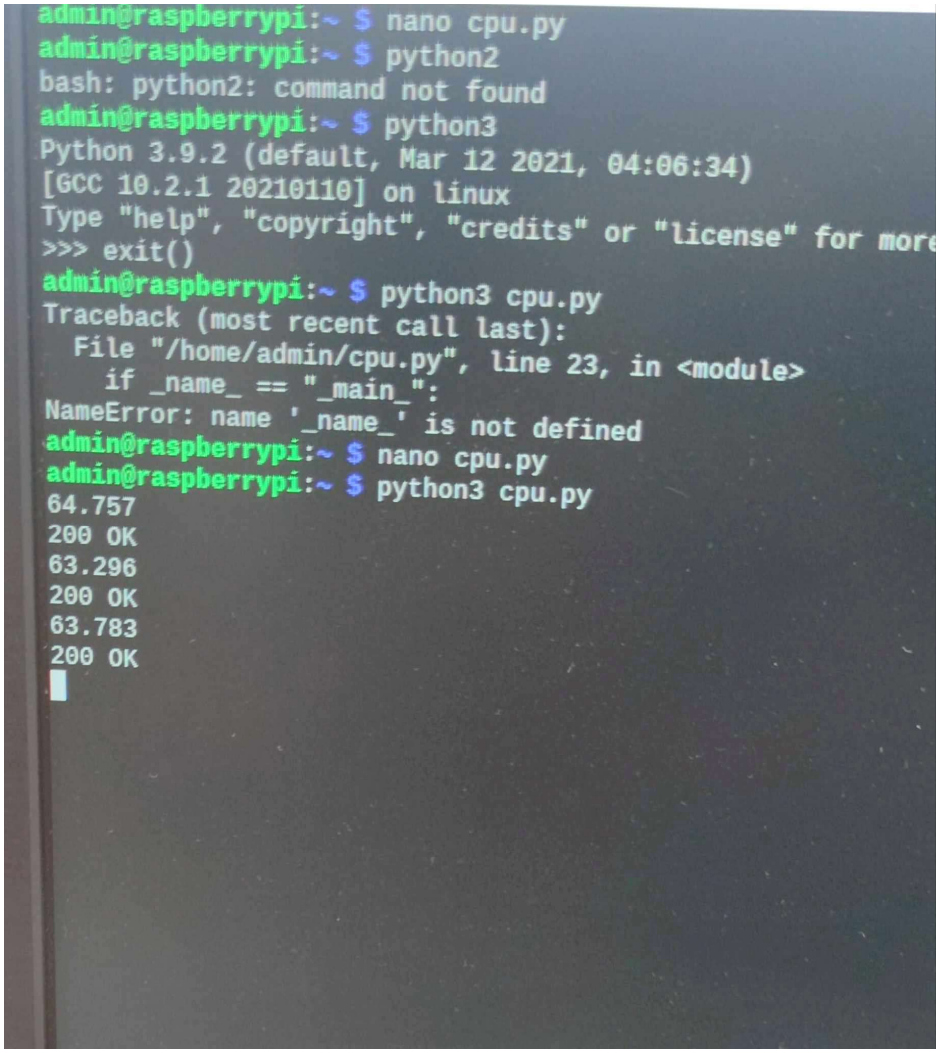
3. Lab 4 Extra

```
import http.client
import urllib.parse
import time

key = "JCQN08TPWH80CNML"  # Put your API Key here

def thermometer():
    # Calculate CPU temperature of Raspberry Pi in Degrees C
    temp = int(open('/sys/class/thermal/thermal_zone0/temp').read()) / 1e3  # Get
Raspberry Pi CPU temp
    params = urllib.parse.urlencode({'field1': temp, 'key': key})
    headers = {"Content-Type": "application/x-www-form-urlencoded", "Accept":
"text/plain"}
    conn = http.client.HTTPConnection("api.thingspeak.com:80")
    try:
        conn.request("POST", "/update", params, headers)
        response = conn.getresponse()
        print(temp)
        print(response.status, response.reason)
        data = response.read()
        conn.close()
    except Exception as e:
        print("Connection failed:", e)

while True:
        thermometer()
        time.sleep(15)  # Adjust the sleep time according to your requirements
```



Channel Stats

Created: about 4 hours ago
Last entry: about 2 hours ago
Entries: 4