IA3

Wednesday, 1 September, 2021 23:43

IA3 31. august

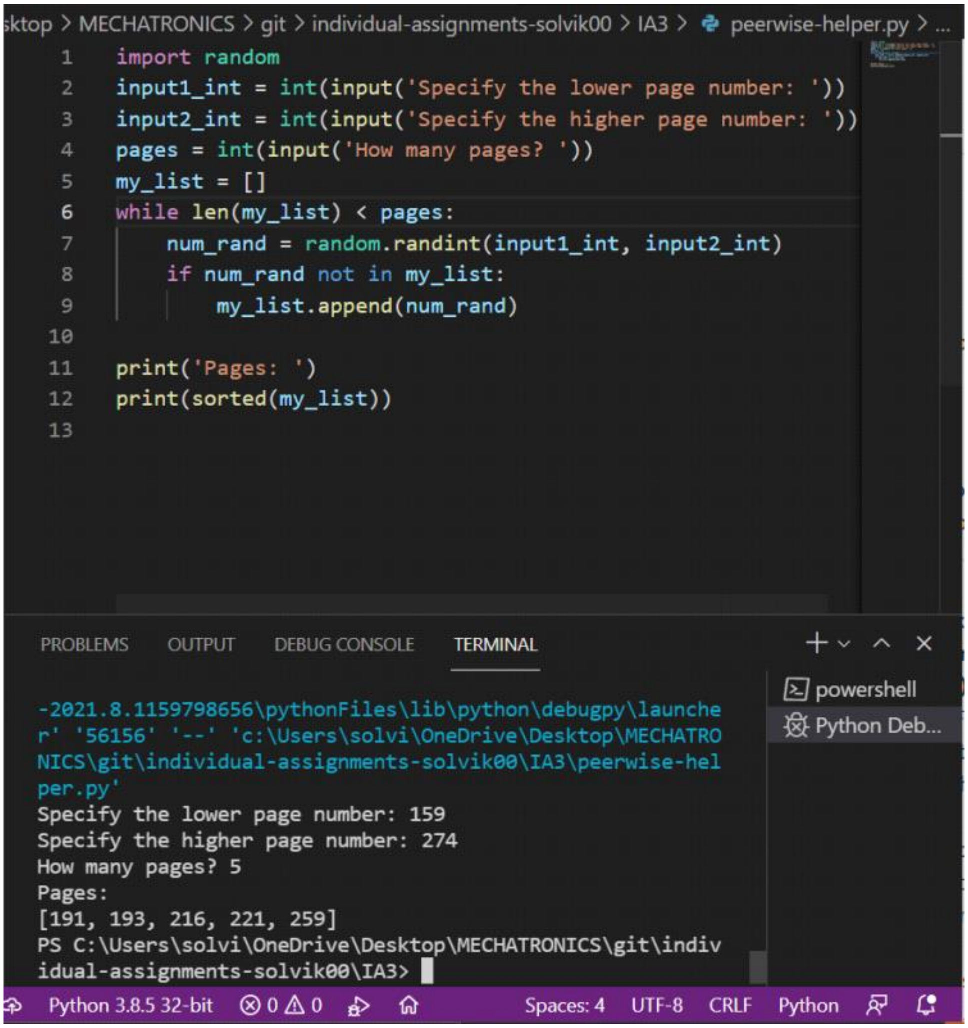
Task 1

Changed "peerwise-helper.py" to choose 5 numbers and that those number won't be the same

Task 2

Created 5 questions on peerwise using "peerwise-helper_2.py"

The questions were from pages 191, 193, 216, 221 and 259.



sktop > MECHATRONICS > git > individual-assignments-solvik00 > IA3 > peerwise-helper.py > ...

```python
import random
input1_int = int(input('Specify the lower page number: '))
input2_int = int(input('Specify the higher page number: '))
pages = int(input('How many pages? '))
my_list = []
while len(my_list) < pages:
    num_rand = random.randint(input1_int, input2_int)
    if num_rand not in my_list:
        my_list.append(num_rand)

print('Pages: ')
print(sorted(my_list))

```

PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
powershell
Python Deb...

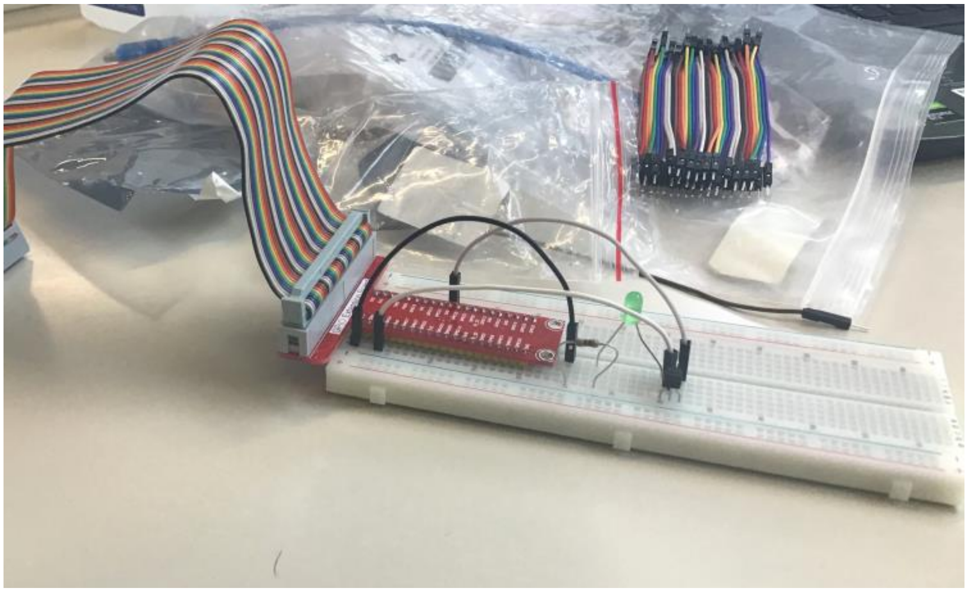
```
-2021.8.1159798656\pythonFiles\lib\python\debugpy\launche
r' '56156' '--' 'c:\Users\solvi\OneDrive\Desktop\MECHATRO
NICS\git\individual-assignments-solvik00\IA3\peerwise-hel
per.py'
Specify the lower page number: 159
Specify the higher page number: 274
How many pages? 5
Pages:
[191, 193, 216, 221, 259]
PS C:\Users\solvi\OneDrive\Desktop\MECHATRONICS\git\indiv
idual-assignments-solvik00\IA3> 
```

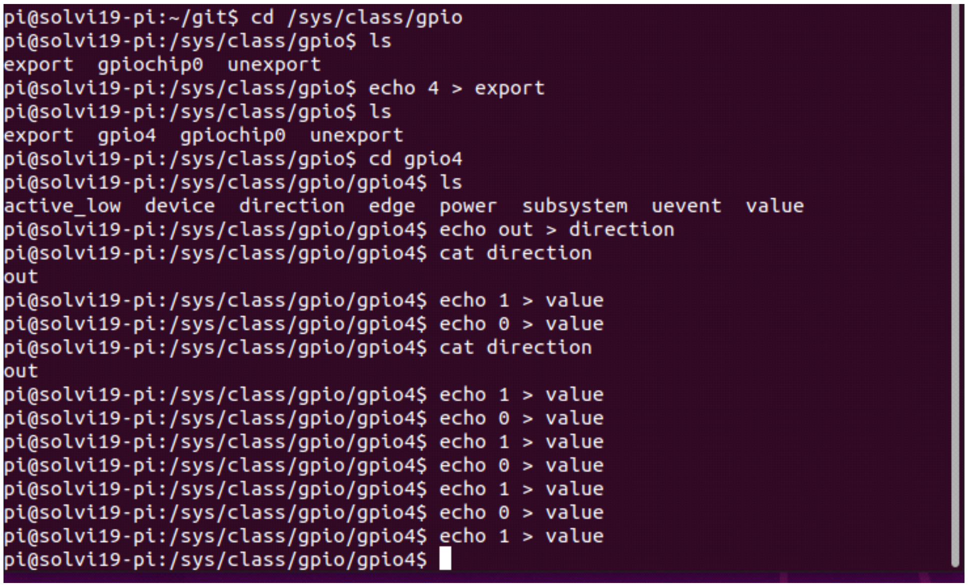
Task 3

Answered 10 questions on Peerwise.

Task 4

Using the Pi Cobbler, a breadboard, a ribbon cable and an LED I made the circuit seen in Fig 8.1 in "Exploring Raspberry Pi".



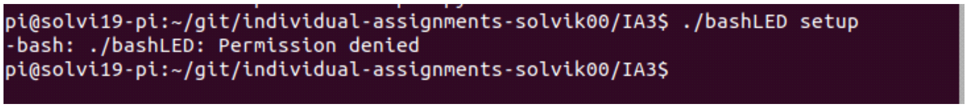


```
pi@solvi19-pi:~/git$ cd /sys/class/gpio
pi@solvi19-pi:/sys/class/gpio$ ls
export  gpiochip0  unexport
pi@solvi19-pi:/sys/class/gpio$ echo 4 > export
pi@solvi19-pi:/sys/class/gpio$ ls
export  gpio4  gpiochip0  unexport
pi@solvi19-pi:/sys/class/gpio$ cd gpio4
pi@solvi19-pi:/sys/class/gpio/gpio4$ ls
active_low  device  direction  edge  power  subsystem  uevent  value
pi@solvi19-pi:/sys/class/gpio/gpio4$ echo out > direction
pi@solvi19-pi:/sys/class/gpio/gpio4$ cat direction
out
pi@solvi19-pi:/sys/class/gpio/gpio4$ echo 1 > value
pi@solvi19-pi:/sys/class/gpio/gpio4$ echo 0 > value
pi@solvi19-pi:/sys/class/gpio/gpio4$ cat direction
out
pi@solvi19-pi:/sys/class/gpio/gpio4$ echo 1 > value
pi@solvi19-pi:/sys/class/gpio/gpio4$ echo 0 > value
pi@solvi19-pi:/sys/class/gpio/gpio4$ echo 1 > value
pi@solvi19-pi:/sys/class/gpio/gpio4$ echo 0 > value
pi@solvi19-pi:/sys/class/gpio/gpio4$ echo 1 > value
pi@solvi19-pi:/sys/class/gpio/gpio4$ echo 0 > value
pi@solvi19-pi:/sys/class/gpio/gpio4$ echo 1 > value
pi@solvi19-pi:/sys/class/gpio/gpio4$ 
```

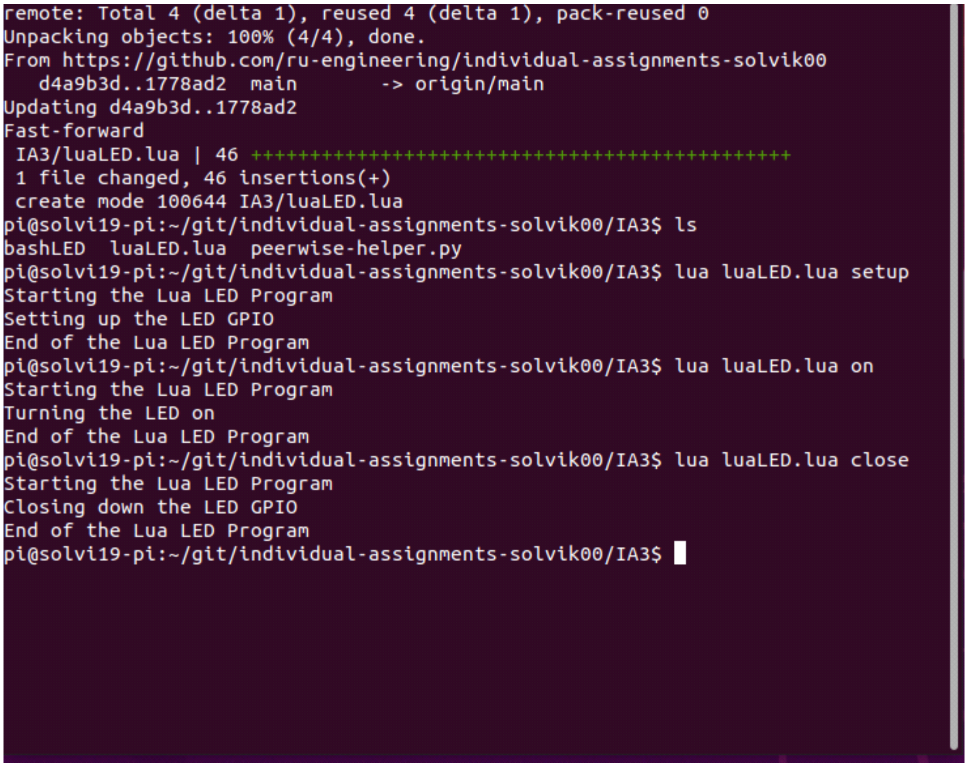
Task 5

Tried to turn the LED on and off using Bash and Lua.

However Bash did not work.



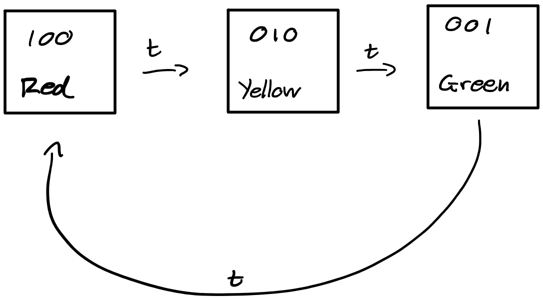
```
pi@solvi19-pi:~/git/individual-assignments-solvik00/IA3$ ./bashLED setup
-bash: ./bashLED: Permission denied
pi@solvi19-pi:~/git/individual-assignments-solvik00/IA3$
```



```
remote: Total 4 (delta 1), reused 4 (delta 1), pack-reused 0
Unpacking objects: 100% (4/4), done.
From https://github.com/ru-engineering/individual-assignments-solvik00
   d4a9b3d..1778ad2  main       -> origin/main
Updating d4a9b3d..1778ad2
Fast-forward
 IA3/luaLED.lua | 46 ++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 46 insertions(+)
 create mode 100644 IA3/luaLED.lua
pi@solvi19-pi:~/git/individual-assignments-solvik00/IA3$ ls
bashLED  luaLED.lua  peerwise-helper.py
pi@solvi19-pi:~/git/individual-assignments-solvik00/IA3$ lua luaLED.lua setup
Starting the Lua LED Program
Setting up the LED GPIO
End of the Lua LED Program
pi@solvi19-pi:~/git/individual-assignments-solvik00/IA3$ lua luaLED.lua on
Starting the Lua LED Program
Turning the LED on
End of the Lua LED Program
pi@solvi19-pi:~/git/individual-assignments-solvik00/IA3$ lua luaLED.lua close
Starting the Lua LED Program
Closing down the LED GPIO
End of the Lua LED Program
pi@solvi19-pi:~/git/individual-assignments-solvik00/IA3$ 
```

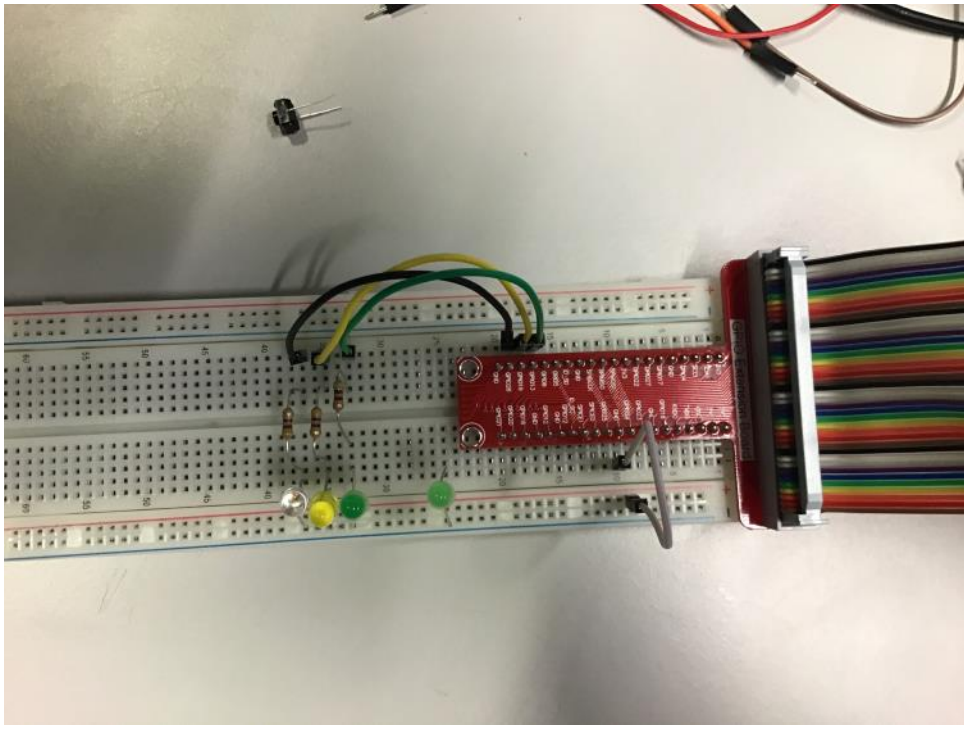
Task 6.

A finite state diagram for a traffic light that goes from Red → Yellow → Green → Red ...



Task 7

Instead of just programming an emulator a python program was made that uses three gpio ports to turn on 3 LED's using gpizero called "stoplight-fsm.py".

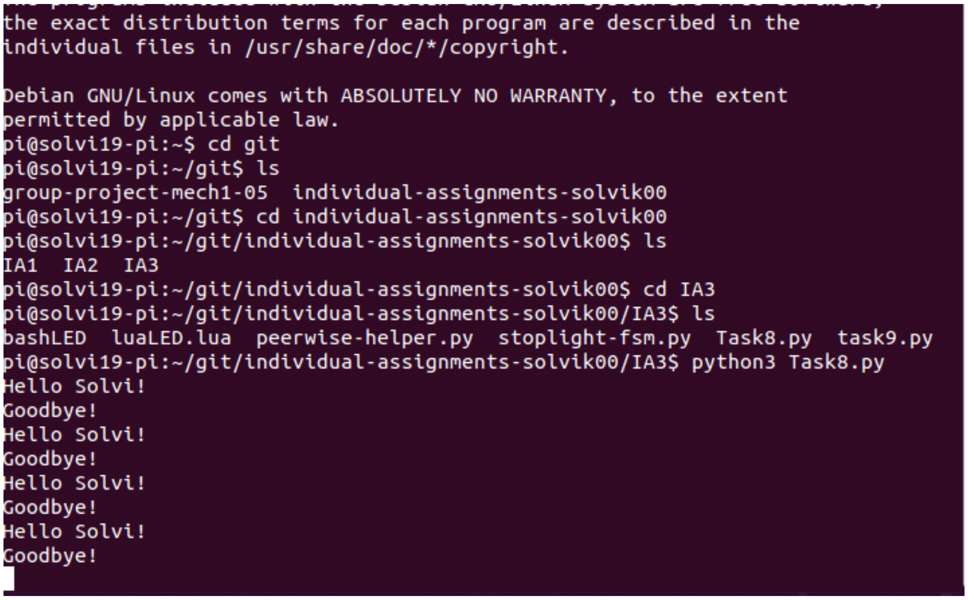


```
from gpiozero import TrafficLights
from time import sleep
from signal import pause
gpio_red = 4
gpio_yellow = 5
gpio_green = 13
lights = TrafficLights(gpio_red, gpio_yellow, gpio_green)
def traffic_light_sequence():
    while True:
        yield (0, 0, 1) # green
        sleep(10)
        yield (0, 1, 0) # amber
        sleep(1)
        yield (1, 0, 0) # red
        sleep(10)
        yield (1, 1, 0) # red+amber
        sleep(1)
lights.source = traffic_light_sequence()
pause()
```

Task 8

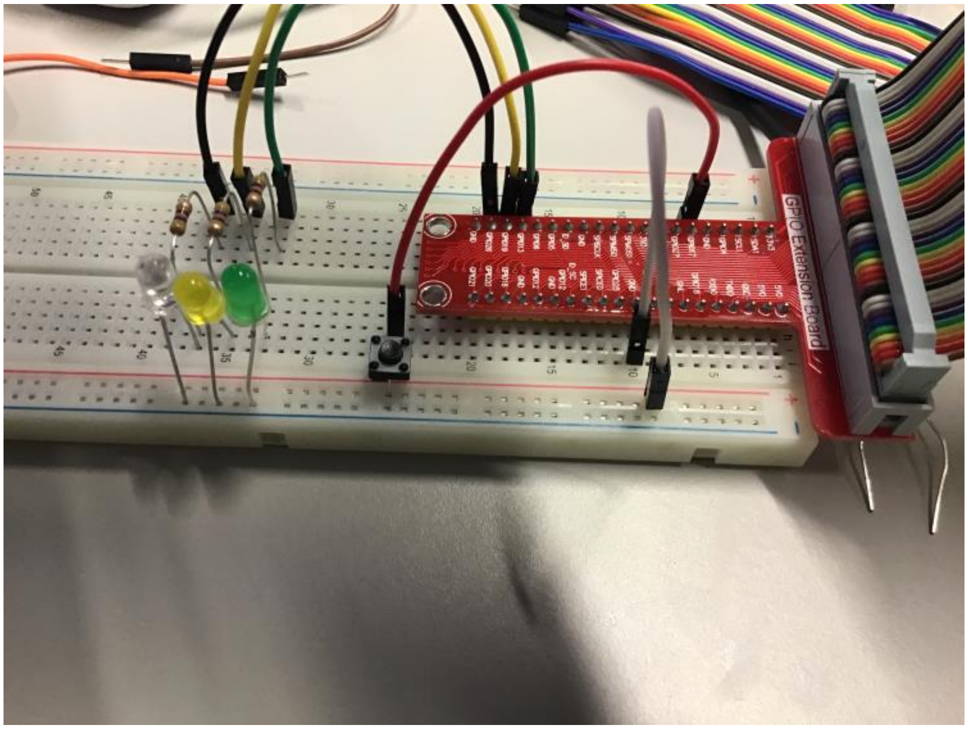
A python program called "Task8.py"

A python program called "Task8.py" was created that said "hello Solvi" when a button, connected to a gpio, was pressed and "Good bye" when it was released.



```
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@solvi19-pi:~$ cd git
pi@solvi19-pi:~/git$ ls
group-project-mech1-05  individual-assignments-solvik00
pi@solvi19-pi:~/git$ cd individual-assignments-solvik00
pi@solvi19-pi:~/git/individual-assignments-solvik00$ ls
IA1  IA2  IA3
pi@solvi19-pi:~/git/individual-assignments-solvik00$ cd IA3
pi@solvi19-pi:~/git/individual-assignments-solvik00/IA3$ ls
bashLED  luaLED.lua  peerwise-helper.py  stoplight-fsm.py  Task8.py  task9.py
pi@solvi19-pi:~/git/individual-assignments-solvik00/IA3$ python3 Task8.py
Hello Solvi!
Goodbye!
Hello Solvi!
Goodbye!
Hello Solvi!
Goodbye!
Hello Solvi!
Goodbye!
```



Task 9

After many hours I gave up on task 9.

```
from gpiozero import PWMLED, Button;
from time import sleep
from signal import pause

led = PWMLED(12)
button = Button(6)
def changerate():
    a = 0.5
    b = 0.5
    while True:
        button.when_pressed = back_again
        while b > 0.1:
            led.on()
            sleep(a)
            led.off()
            sleep(b)
            a += 0.1
            b -= 0.1
            button.when_pressed = back_again
        while a != b:
            led.on()
            sleep(a)
            led.off()
            sleep(b)
            a -= 0.1
            b += 0.1
            button.when_pressed = back_again

def back_again():
    led.blink(0.5,0.5,0,0,None,True)
    button.when_pressed = changerate
back_again()



pause()
```

Managed to get it working.
Just a misunderstanding
about how PWM worked.

```
from gpiozero import PWMLED, Button;
from time import sleep
from signal import pause
led_gpio = 12
button_gpio = 6
led = PWMLED(led_gpio)
button = Button(button_gpio)
switch = False
```

```
led.value=0.5
def changerate():
    global switch
    if switch == True:
        switch = False
    else:
        switch = True
while True:
    button.when_pressed = changerate
    if switch == True:
        for i in range(50,101):
            led.value = i / 100
            sleep(0.01)
        for i in range(100,51,-1):
            led.value = i / 100
            sleep(0.01)
    else:
        led.value = 0.5
```