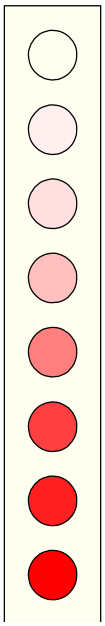
ECE 476/676 - Homework #11

WiFi - Due Monday, November 25th

Starter Tree



Write a Python program so you can control a starter tree with your cell phone over a WiFi network.

- A NeoPixel acts as the starter tree (0 to 8 lights on)
- The Pico updates a web page so that you can see the status of the LEDs on your cell phone in real time.
- Your cell phone can send commands to the PICO to
  - Clear the lights (new race) and
  - Start a race (start turning on the lights)

Problem 1) NeoPixel: Write a Pyton program so that the Pico controls the lights of a starter tree with push buttons:

- GP14: Clear the LEDs
- GP15: Start a race

When a race is started,

- The LEDs turn on one at a time,
- Each LED turns on once every second
- Once all eight LEDs are on, they stay on (race has begun)

Verify your code works.

Planning ahead, use interrupt to drive the starter tree. Two global variables are used:

- Run:
  - 1: Race is starting. Increment N every T seconds until you get to 16
  - 0: Race is waiting. All lights are off
- N:
  - 0: All lights are off
  - 1-15: Lights are red (should be yellow): ready, set...
  - 16: All lights are green (go!)

Timer Interrupt: Called every 500ms (with 16 LEDs, 8 seconds to start a race)

- Run = 1 starts the race
- When Run == 1:
  - N increments up to 16
  - When N == 16, the race is going
- When Run == 0:
  - Turn off the lights
  - Wait until Run == 1

```
#---- Timer Interrupt ---------
tim = Timer()
def tic(timer):
    global N, Run
    if(Run):
        N = min(N+1, 16)
        if(N < 16):
            np[N-1] = (50, 0, 0)
            LCD.Text2('Ready   ', 20, 100, White, Black)
            LCD.Text2(str(N) + '  ', 20, 130, White, Black)
        else:
            np.fill([0,50,0])
            LCD.Text2('Go!     ', 20, 100, White, Black)
            LCD.Text2(str(N) + '  ', 20, 130, White, Black)

    else:
        N = 0
        np.fill([0,0,0])
        LCD.Text2('Waiting    ', 20, 100, White, Black)
        LCD.Text2(str(N) + '  ', 20, 130, White, Black)
    np.write()


tim.init(period = 500, mode=Timer.PERIODIC, callback=tic)
```

Edge Interrupts

- GP15: Starts the race
- GP14: Clears the lights

```
# ---- Edge Interrupts ---------

p15 = Pin(15,Pin.IN,Pin.PULL_UP)
def Start(pin):
    global N, Run
    Run = 1
    N = 0

p15.irq(trigger=Pin.IRQ_FALLING, handler=Start)

p14 = Pin(14,Pin.IN,Pin.PULL_UP)
def Stop(pin):
    global N, Run
    Run = 0
    N = 0

p14.irq(trigger=Pin.IRQ_FALLING, handler=Stop)
```

Main Loop: Doesn't really do anything but display the status

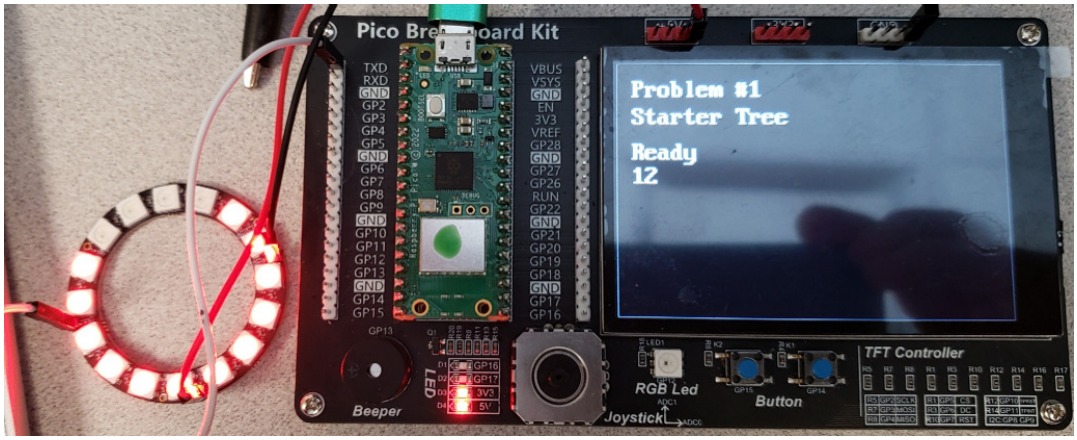
```
while(1):
    print(N, Run)
    sleep(0.2)
```

Shell Window:

```
N   Run
0   0
0   0
1   1     <<  button was pressed
2   1
3   1
4   1
```

Testing:

- GP15 starts the race (the lights start turning non, shows up on the LCD display)
- When all 16 lights are on, the lights turn green
- GP14 turns off the lights

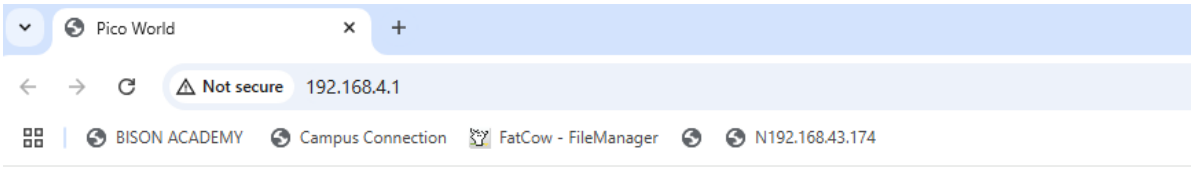


Problem 2) Pico to Cell Phone over WiFi: Set up a web page so that you can see the status of the starter tree on your cell phone. It's your choice if you set up the PIC as a host or a client.

Verify that you can see the status of the starter tree in real time on your cell phone.

For this problem, use AP mode and a table for the web page. Modify the table html code from lecture #33 to display the status and N:

```
<!DOCTYPE html><html>
<head>
  <title>Pico World</title>
  <meta http-equiv="refresh" content="1">
  <style>
    table {  border-collapse: collapse;   width: 80%; }
    th, td {  text-align: center;  padding: 8px; }
    tr:nth-child(even) { background-color: #EEDDDD; }
    </style>
  </head>
<body>
  <h2>Starter  Tree</h2>
  <p>ECE 476 HW #11</p>
  <table>
    <tr>   <th>Parameter</th>   <th>Value</th>    </tr>
    <tr>   <td>Status</td>   <td> aaaaa </td>     </tr>
    <tr>   <td>N</td>   <td> bbbbb </td>      </tr>
    </table>
  </body>
</html>
```



Starter Tree

ECE 476 HW #11

Parameter	Value
Status	Yellow
N	6

To update the status (red - yellow - green) and the number of LEDs lit up, pass that data to the Web_Page() routine

```
def web_page(x0, x1):
    f = open("HW11.html","rt")
    x = f.readlines()
    y = x[0]
    for i in range(0,len(x)):
        y = y + x[i]
    y = y.replace('\r\n',' ')
    y = y.replace('aaaaa', x0)
    y = y.replace('bbbbb', x1)
    return(y)
```

In the main routine, turn on the WiFi in AP mode:

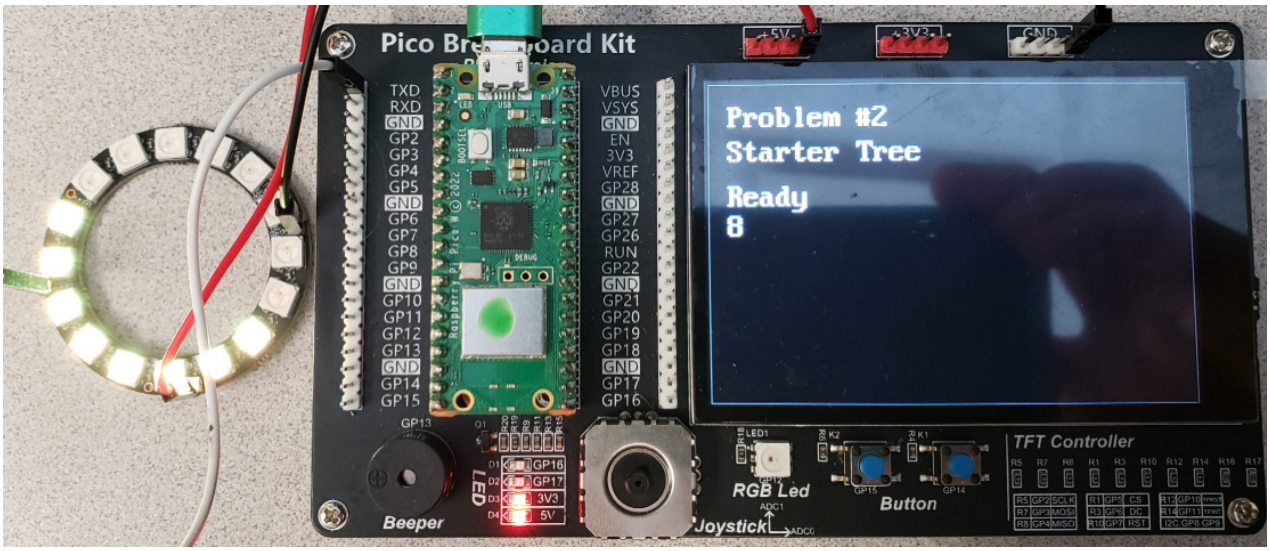
```
# --- Start of main loop ------------

ssid = 'Pico-Network'
password = 'PASSWORD'

ap = network.WLAN(network.AP_IF)
ap.config(essid=ssid, password=password)
ap.active(True)

while ap.active() == False:
    pass
print('AP Mode Is Active, You can Now Connect')
print('IP Address To Connect to:: ' + ap.ifconfig()[0])

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('', 80))
s.listen(5)
```

In the main loop (which doesn't really do anything - as per problem #1), keep updating the web page:

```
while(1):
    conn, addr = s.accept()
    print('conn = ', conn)
    print('addr = ', addr)
    print('Got a connection from %s' % str(addr))
    request = conn.recv(1024)
    #print('Content = %s' % str(request))
    if(N == 0):
        response = web_page('Red', 'Waiting')
    elif(N < 16):
        response = web_page('Yellow', str(N))
    else:
        response = web_page('Green', 'Go!')
    conn.send(response)
    conn.close()
```

Validation:

- The buttons control the starter tree (GP15 = start, GP14 = clear)
- You can see the status of the starter tree on the web page
- You can see the status of the starter tree on the LCD display



Problem 3) Cell Phone to Pico over WiFi: Add to the previous design a way for you to use your cell phone and the WiFi network to

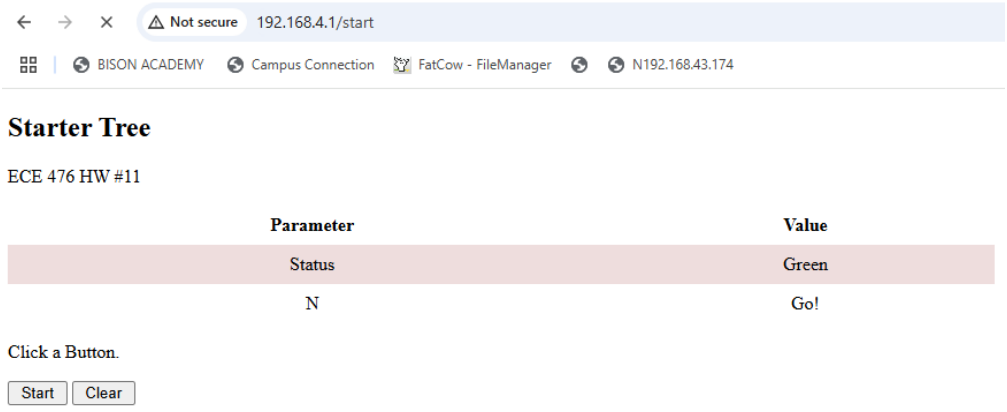
- Clear the starter tree (replacing button GP14), and
- Start a race (replacing button GP15)

Verify your code works.

Modify the previous HTML code to add push puttons with hyperlinks, The hyperlinks return the messages

- start
- clear

```
<!DOCTYPE html><html>
<head>
  <title>Pico World</title>
  <meta http-equiv="refresh" content="1">
  <style>
    table {  border-collapse: collapse;   width: 80%; }
    th, td {  text-align: center;  padding: 8px; }
    tr:nth-child(even) { background-color: #EEDDDD; }
    </style>
  </head>
<body>
  <h2>Starter  Tree</h2>
  <p>ECE 476 HW #11</p>
  <table>
    <tr>   <th>Parameter</th>   <th>Value</th>    </tr>
    <tr>   <td>Status</td>   <td> bbbbb </td>     </tr>
    <tr>   <td>N</td>   <td> ccccc </td>      </tr>
    </table>

<p>Click a Button.</p>
<form>
<p><a href="http://aaaaa/start"><input type="button" value=" Start
"></a>
<a href="http://aaaaa/clear"><input type="button" value=" Clear "></a>
</p>
</form>

  </body>
</html>
```



With hyperlinks, the *Web_Page()* routine now needs to know the IP address of the Pico:

```
def web_page(IP_Address, x0, x1):
    f = open("HW11_p3.html","rt")
    x = f.readlines()
    y = x[0]
    for i in range(0,len(x)):
        y = y + x[i]
    y = y.replace('\r\n',' ')
    y = y.replace('aaaaa', IP_Address)
    y = y.replace('bbbbb', x0)
    y = y.replace('ccccc', x1)
    return(y)
```

In the main loop, now look at the message that's sent from the client

- Keep checking until you receive a message with *favicon*
- Each ping, reply with an update of the starter tree status

```
while(1):
    flag = 0
    while(flag == 0):
        conn, addr = s.accept()
        request = conn.recv(1024)
        request = request.decode('utf-8')
        if(request.find('favicon') > 0):
            flag = 1
        else:
            if(N == 0):
                response = web_page(IP_Address, 'Red', 'Waiting')
            elif(N < 16):
                response = web_page(IP_Address, 'Yellow', str(N))
            else:
                response = web_page(IP_Address, 'Green', 'Go!')
            conn.send(response)
            conn.close()
```

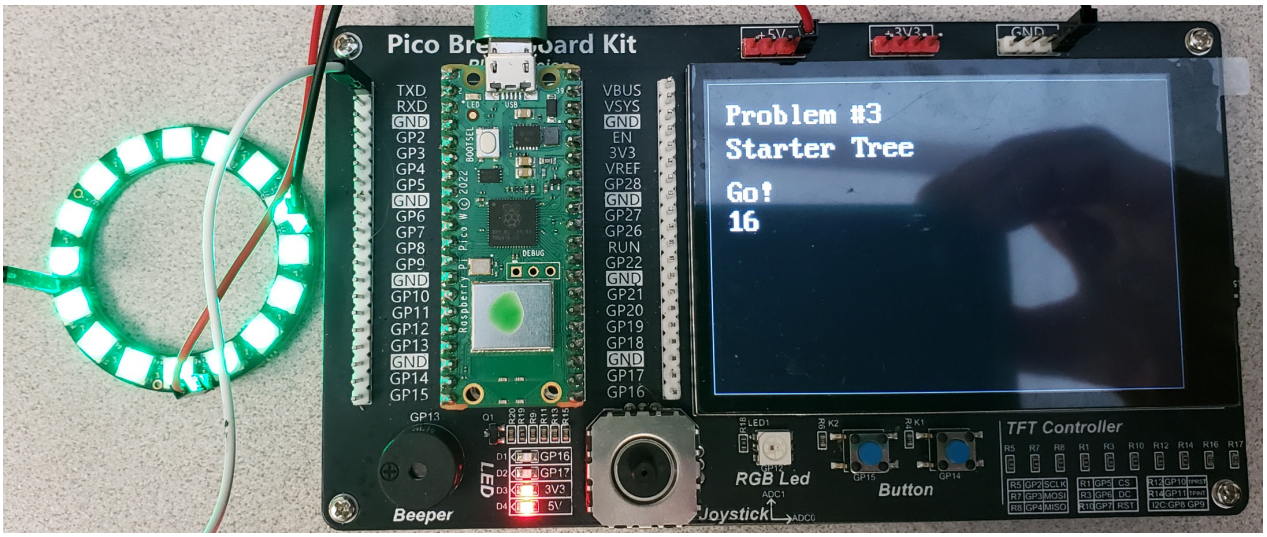
When you receive a ping with a *favicon* message,

- Pull out the mesage
- Check if it's *start* or *clear*
- Update the global variables (Run, N) accordingly

```
n = request.find('Referer:')+9
request = request[n:]
n = request.find('\r\n')
request = request[0:n]
print('-'*10)
print(request)
for i in range(0,10):
    n = request.find('/')+1
    if(n>0):
        request = request[n:]
print(request)
if(request == 'start'):
    if(Run == 0):
        N = 0
        np.fill([0,0,0])
    Run = 1
if(request == 'clear'):
    if(Run == 1):
        N = 0
    Run = 0
if(N == 0):
    response = web_page(IP_Address, 'Red', 'Waiting')
elif(N < 16):
    response = web_page(IP_Address, 'Yellow', str(N))
else:
    response = web_page(IP_Address, 'Green', 'Go!')
conn.send(response)
conn.close()
```

Check:

- Able to start and clear the starter tree using GP15 and GP14
- Able to start and clear the starter tree using hyperlink buttons from the client
- Able to see the status of the starter tree on
  - the NeoPixels
  - the LCD display
  - the client's web page



Problem 4) Demo your starter tree

- Video preferred