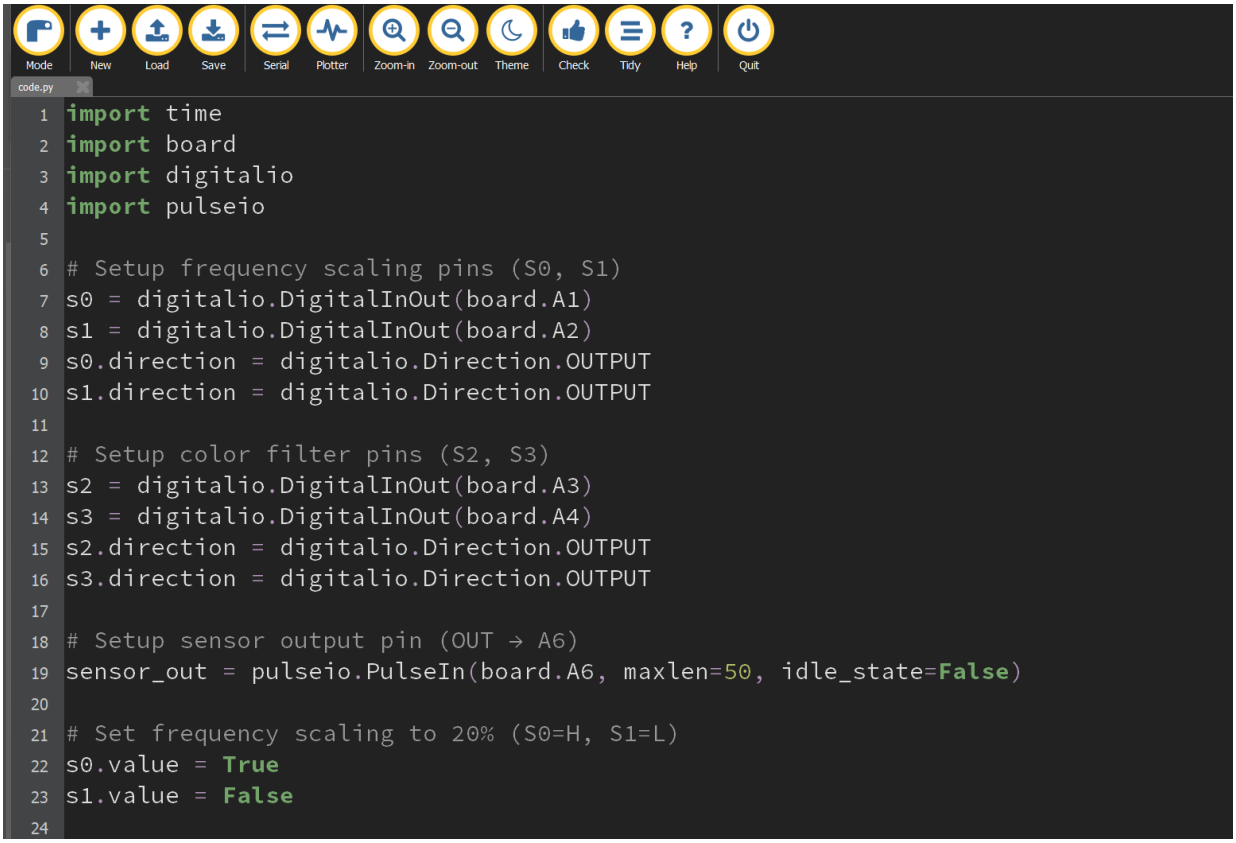
Coding part




```python
import time
import board
import digitalio
import pulseio

# Setup frequency scaling pins (S0, S1)
s0 = digitalio.DigitalInOut(board.A1)
s1 = digitalio.DigitalInOut(board.A2)
s0.direction = digitalio.Direction.OUTPUT
s1.direction = digitalio.Direction.OUTPUT

# Setup color filter pins (S2, S3)
s2 = digitalio.DigitalInOut(board.A3)
s3 = digitalio.DigitalInOut(board.A4)
s2.direction = digitalio.Direction.OUTPUT
s3.direction = digitalio.Direction.OUTPUT

# Setup sensor output pin (OUT → A6)
sensor_out = pulseio.PulseIn(board.A6, maxlen=50, idle_state=False)

# Set frequency scaling to 20% (S0=H, S1=L)
s0.value = True
s1.value = False

```

```python
def read_color(s2_state, s3_state):
    s2.value = s2_state
    s3.value = s3_state
    time.sleep(0.1)    # let filter settle

    sensor_out.clear()
    sensor_out.resume()
    time.sleep(0.05)  # collect pulses
    sensor_out.pause()

    n = len(sensor_out)
    if n == 0:
        return 0

    # Directly sum the values inside PulseIn
    total = 0
    for i in range(n):
        total += sensor_out[i]

    avg_pulse = total / n
    freq = 1_000_000 / avg_pulse  # Hz
    return int(freq)

```

```
def classify_color(r, g, b, c):
    total = r + g + b
    if total == 0:
        return "None"

    rn, gn, bn = r / total, g / total, b / total

    # Debug print
    print("Norm → R:", round(rn, 2), "G:", round(gn, 2), "B:", round(bn, 2))

    if rn > 0.40 and rn > gn + 0.10 and rn > bn + 0.10:
        return "Red"
    elif gn > 0.35 and gn > rn + 0.05 and gn > bn + 0.05:
        return "Green"
    elif bn > 0.35 and bn > rn + 0.05 and bn > gn + 0.05:
        return "Blue"
    elif abs(rn - gn) < 0.05 and abs(rn - bn) < 0.05:
        return "White"
    else:
        return "Unknown"

while True:
    red   = read_color(False, False)   # Red
    blue  = read_color(False, True)    # Blue
```

```
    green = read_color(True, True)     # Green
    clear = read_color(True, False)    # Clear

    color = classify_color(red, green, blue, clear)
    print("R:", red, "G:", green, "B:", blue, "C:", clear, "→", color)
    time.sleep(0.5)
```

You can adjust the value of checking condition and modify and see the output.