```
 1 import RPi.GPIO as GPIO
 2 import fileWriter
 3
 4 def turn_off(led):
 5     if GPIO.input(led):
 6         print("Turn off: " + str(led))
 7         GPIO.output(led, False)
 8
 9
10 def turn_on(led):
11     if not GPIO.input(led):
12         print("Turn on: " + str(led))
13         GPIO.output(led, True)
14
15
16 class Button:
17
18     def __init__(self, name, button_id, led1, led2,
   led3):
19         self.name = name
20         self.buttonId = button_id
21         self.led1 = led1
22         self.led2 = led2
23         self.led3 = led3
24         self.isRunning = False
25         self.didPress = False
26         self.holdLength = 0
27         self.state = 0
28         self.HOLD_THRESHOLD = 16
29
30     def setup(self):
31         if self.isRunning:
32             return
33
34         self.isRunning = True
35
36         GPIO.setup(self.led1, GPIO.OUT)
37         GPIO.setup(self.led2, GPIO.OUT)
38         GPIO.setup(self.led3, GPIO.OUT)
39
40         GPIO.setup(self.buttonId, GPIO.IN,
```

```
40 pull_up_down=GPIO.PUD_DOWN)
41
42         turn_off(self.led1)
43         turn_off(self.led2)
44         turn_off(self.led3)
45
46     def check_state(self):
47
48         if GPIO.input(self.buttonId):
49             if not self.didPress:
50                 self.didPress = True
51                 self.holdLength = 0
52                 print(self.name + " pressed, moving
   to state: " + str(self.state))
53         elif self.didPress:
54             if self.holdLength != -1:
55                 self.state += 1
56             self.didPress = False
57             self.holdLength = 0
58             print(self.name + " released")
59
60         #  Increment the holdLength if the button is
   held
61         if self.didPress and self.holdLength != -1:
62             self.holdLength += 1
63
64         #  Check if we have been holding longer than
   the threshold
65         if self.holdLength >= self.HOLD_THRESHOLD:
66             # Write the amount of pills taken
67             fileWriter.save_to_file(self.name, self.
   state)
68             self.holdLength = -1
69             self.state = 0
70             print(self.name + " reset by holding")
71
72         #  Turn on/off leds based on state
73         if self.state >= 3:
74             turn_on(self.led3)
75             self.state = 3
76
```

```
77          if self.state >= 2:
78              turn_on(self.led2)
79
80          if self.state >= 1:
81              turn_on(self.led1)
82
83          if self.state == 0:
84              turn_off(self.led1)
85              turn_off(self.led2)
86              turn_off(self.led3)
87
88      def get_name(self):
89          return self.name
90
```

```
 1 Program Started 16:29:55 2022-11-22
 2
 3 16:29:55 2022-11-22 :: Red 2
 4 16:29:55 2022-11-22 :: Yellow 1
 5 16:29:55 2022-11-22 :: Green 0
 6 16:29:55 2022-11-22 :: Blue 0
 7 16:29:55 2022-11-22 :: Red 2
 8 16:29:55 2022-11-22 :: Yellow 2
 9 16:29:55 2022-11-22 :: Green 3
10 16:29:55 2022-11-22 :: Blue 1
11 16:29:55 2022-11-22 :: Red 0
12 16:29:55 2022-11-22 :: Yellow 1
13 16:29:55 2022-11-22 :: Green 0
14 16:29:55 2022-11-22 :: Blue 2
15 16:29:55 2022-11-22 :: Red 2
16 16:29:55 2022-11-22 :: Yellow 1
17 16:29:55 2022-11-22 :: Green 1
18 16:29:55 2022-11-22 :: Blue 3
19 16:29:55 2022-11-22 :: Red 1
20 16:29:55 2022-11-22 :: Yellow 3
21 16:29:55 2022-11-22 :: Green 0
22 16:29:55 2022-11-22 :: Blue 0
23 16:29:55 2022-11-22 :: Red 0
24 16:29:55 2022-11-22 :: Yellow 0
25 16:29:55 2022-11-22 :: Green 0
26 16:29:55 2022-11-22 :: Blue 2
27 Program Started 16:29:55 2022-11-22
28
29 Program Started 16:29:55 2022-11-22
30
31 16:29:55 2022-11-22 :: Red 3
32 16:29:55 2022-11-22 :: Yellow 3
33 16:29:55 2022-11-22 :: Green 0
34 16:29:55 2022-11-22 :: Blue 3
35 16:29:55 2022-11-22 :: Red 0
36 16:29:55 2022-11-22 :: Yellow 2
37 16:29:55 2022-11-22 :: Green 3
38 16:29:55 2022-11-22 :: Blue 0
39 16:29:55 2022-11-22 :: Red 0
40 16:29:55 2022-11-22 :: Yellow 3
41 16:29:55 2022-11-22 :: Green 1
```

```
42 16:29:55 2022-11-22 :: Blue 0
43 Program Started 16:29:55 2022-11-22
44
45 16:29:55 2022-11-22 :: Red 2
46 16:29:55 2022-11-22 :: Yellow 1
47 16:29:55 2022-11-22 :: Green 1
48 16:29:55 2022-11-22 :: Blue 2
49 16:29:55 2022-11-22 :: Red 3
50 16:29:55 2022-11-22 :: Yellow 0
51 16:29:55 2022-11-22 :: Green 0
52 16:29:55 2022-11-22 :: Blue 3
53 16:29:55 2022-11-22 :: Red 3
54 16:29:55 2022-11-22 :: Yellow 0
55 16:29:55 2022-11-22 :: Green 3
56 16:29:55 2022-11-22 :: Blue 0
57 16:29:55 2022-11-22 :: Red 1
58 16:29:55 2022-11-22 :: Yellow 0
59 16:29:55 2022-11-22 :: Green 3
60 16:29:55 2022-11-22 :: Blue 1
61 16:29:55 2022-11-22 :: Red 0
62 16:29:55 2022-11-22 :: Yellow 1
63 16:29:55 2022-11-22 :: Green 3
64 16:29:55 2022-11-22 :: Blue 1
65 16:29:55 2022-11-22 :: Red 3
66 16:29:55 2022-11-22 :: Yellow 3
67 16:29:55 2022-11-22 :: Green 3
68 16:29:55 2022-11-22 :: Blue 2
69 Program Started 16:29:55 2022-11-22
70
71 16:29:55 2022-11-22 :: Red 2
72 16:29:55 2022-11-22 :: Yellow 2
73 16:29:55 2022-11-22 :: Green 2
74 16:29:55 2022-11-22 :: Blue 0
75 16:29:55 2022-11-22 :: Red 2
76 16:29:55 2022-11-22 :: Yellow 0
77 16:29:55 2022-11-22 :: Green 1
78 16:29:55 2022-11-22 :: Blue 2
79 16:29:55 2022-11-22 :: Red 0
80 16:29:55 2022-11-22 :: Yellow 3
81 16:29:55 2022-11-22 :: Green 3
82 16:29:55 2022-11-22 :: Blue 2
```

```
 83 16:29:55 2022-11-22 :: Red 0
 84 16:29:55 2022-11-22 :: Yellow 2
 85 16:29:55 2022-11-22 :: Green 1
 86 16:29:55 2022-11-22 :: Blue 2
 87 16:29:55 2022-11-22 :: Red 1
 88 16:29:55 2022-11-22 :: Yellow 3
 89 16:29:55 2022-11-22 :: Green 3
 90 16:29:55 2022-11-22 :: Blue 3
 91 16:29:55 2022-11-22 :: Red 2
 92 16:29:55 2022-11-22 :: Yellow 3
 93 16:29:55 2022-11-22 :: Green 0
 94 16:29:55 2022-11-22 :: Blue 1
 95 16:29:55 2022-11-22 :: Red 0
 96 16:29:55 2022-11-22 :: Yellow 1
 97 16:29:55 2022-11-22 :: Green 2
 98 16:29:55 2022-11-22 :: Blue 3
 99 16:29:55 2022-11-22 :: Red 2
100 16:29:55 2022-11-22 :: Yellow 2
101 16:29:55 2022-11-22 :: Green 0
102 16:29:55 2022-11-22 :: Blue 2
103 16:29:55 2022-11-22 :: Red 2
104 16:29:55 2022-11-22 :: Yellow 1
105 16:29:55 2022-11-22 :: Green 0
106 16:29:55 2022-11-22 :: Blue 0
107 Program Started 16:29:55 2022-11-22
108
109 16:29:55 2022-11-22 :: Red 1
110 16:29:55 2022-11-22 :: Yellow 2
111 16:29:55 2022-11-22 :: Green 1
112 16:29:55 2022-11-22 :: Blue 2
113 16:29:55 2022-11-22 :: Red 3
114 16:29:55 2022-11-22 :: Yellow 1
115 16:29:55 2022-11-22 :: Green 3
116 16:29:55 2022-11-22 :: Blue 0
117 16:29:55 2022-11-22 :: Red 3
118 16:29:55 2022-11-22 :: Yellow 3
119 16:29:55 2022-11-22 :: Green 0
120 16:29:55 2022-11-22 :: Blue 3
121 16:29:55 2022-11-22 :: Red 3
122 16:29:55 2022-11-22 :: Yellow 0
123 16:29:55 2022-11-22 :: Green 0
```

```
124 16:29:55 2022-11-22 :: Blue 2
125 16:29:55 2022-11-22 :: Red 1
126 16:29:55 2022-11-22 :: Yellow 0
127 16:29:55 2022-11-22 :: Green 1
128 16:29:55 2022-11-22 :: Blue 3
129 16:29:55 2022-11-22 :: Red 3
130 16:29:55 2022-11-22 :: Yellow 1
131 16:29:55 2022-11-22 :: Green 1
132 16:29:55 2022-11-22 :: Blue 1
133 Program Started 16:29:55 2022-11-22
134
135 16:29:55 2022-11-22 :: Red 0
136 16:29:55 2022-11-22 :: Yellow 0
137 16:29:55 2022-11-22 :: Green 1
138 16:29:55 2022-11-22 :: Blue 0
139 16:29:55 2022-11-22 :: Red 0
140 16:29:55 2022-11-22 :: Yellow 3
141 16:29:55 2022-11-22 :: Green 1
142 16:29:55 2022-11-22 :: Blue 0
143 16:29:55 2022-11-22 :: Red 2
144 16:29:55 2022-11-22 :: Yellow 3
145 16:29:55 2022-11-22 :: Green 1
146 16:29:55 2022-11-22 :: Blue 1
147 Program Started 16:29:55 2022-11-22
148
149 Program Started 16:29:55 2022-11-22
150
151 Program Started 16:29:55 2022-11-22
152
153 16:29:55 2022-11-22 :: Red 1
154 16:29:55 2022-11-22 :: Yellow 3
155 16:29:55 2022-11-22 :: Green 0
156 16:29:55 2022-11-22 :: Blue 1
157 16:29:55 2022-11-22 :: Red 2
158 16:29:55 2022-11-22 :: Yellow 0
159 16:29:55 2022-11-22 :: Green 2
160 16:29:55 2022-11-22 :: Blue 3
161 16:29:55 2022-11-22 :: Red 1
162 16:29:55 2022-11-22 :: Yellow 1
163 16:29:55 2022-11-22 :: Green 0
164 16:29:55 2022-11-22 :: Blue 2
```

```
165 16:29:55 2022-11-22 :: Red 1
166 16:29:55 2022-11-22 :: Yellow 2
167 16:29:55 2022-11-22 :: Green 2
168 16:29:55 2022-11-22 :: Blue 1
169 16:29:55 2022-11-22 :: Red 0
170 16:29:55 2022-11-22 :: Yellow 3
171 16:29:55 2022-11-22 :: Green 3
172 16:29:55 2022-11-22 :: Blue 2
173 16:29:55 2022-11-22 :: Red 0
174 16:29:55 2022-11-22 :: Yellow 2
175 16:29:55 2022-11-22 :: Green 0
176 16:29:55 2022-11-22 :: Blue 0
177
```

```python
from datetime import date
from datetime import datetime

output_file = 'output.log'


# saves a pill counters value to file
def save_to_file(light, val):
    file = open(output_file, "a")  # open file in append

    out = str(datetime.now().strftime("%H:%M:%S")) + ' ' + str(date.today()) + ' :: ' + str(light) + ' ' + str(val) + '\n'

    file.write(str(out))  # write out to file
    file.close()  # close file


# appends a star message to the output file
def announce_start():
    file = open(output_file, "a")  # open file in append

    out = 'Program Started ' + str(datetime.now().strftime("%H:%M:%S")) + ' ' + str(date.today()) + '\n\n'

    file.write(str(out))  # write out to file
    file.close()  # close file
```

```python
import RPi.GPIO as GPIO

import fileWriter
from Button import Button
import time

fileWriter.announce_start()

GPIO.setmode(GPIO.BOARD)

blueButton = Button("Blue", 13, 7, 10, 8)
greenButton = Button("Green", 19, 15, 18, 16)
yellowButton = Button("Yellow", 36, 21, 24, 23)
redButton = Button("Red", 31, 40, 29, 26)

blueButton.setup()
greenButton.setup()
yellowButton.setup()
redButton.setup()

while (True):
    blueButton.check_state()
    greenButton.check_state()
    yellowButton.check_state()
    redButton.check_state()
    time.sleep(.125)

```