

```

1  '''
2      /'{}>
3      /'{}>
4      //'-; ;--'\
5      //'-; ;--'\
6      m m m m
7
8  *****
9  *
10 * EET-418
11 * 27 April 2022
12 * Senior Project: Smart package Delivery Box
13 *
14 * The purpose of this program is to act as the BlinkM.py
15 * file for the SPDB project. It is intended to be imported
16 * into the main program.
17 *
18 * Created by: Brandon Empie
19 *
20 *****
21 '''
22
23 import time
24 import smbus2
25 from smbus2 import SMBus, i2c_msg
26 address = 0x09
27 bus = smbus2.SMBus(1) #creating object
28
29
30 #Standard Colors
31 red = [0xff, 0x00, 0x00]
32 green = [0x00, 0xff, 0x00]
33 blue = [0x00, 0x00, 0xff]
34 white = [0xff, 0xff, 0xff]
35 cyan = [0x00, 0xff, 0xff]
36 yellow = [0xff, 0xff, 0x00]
37 purple = [0xff, 0x00, 0xff]
38 orange = [0xff, 0x12, 0x00]
39 off = [0x00, 0x00, 0x00]
40
41 AllColors = (red,white,blue,green,cyan,yellow,purple,orange,off)
42
43 #Special Circumstance Script
44 locator = [7, 0x00, 0x00]
45 full = [3, 0x00, 0x00]
46
47 #Fade to HSB Color
48 pink = [245, 0xff, 0xff]
49
50 def HSBFade(color):
51     bus.open(1) #open i2c bus 1
52     bus.write_i2c_block_data(address, 0x68, color)
53     time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
54     bus.close() #close i2c bus 1
55
56 def SolidColor(color):
57     bus.open(1) #open i2c bus 1
58     bus.write_i2c_block_data(address, 0x6e, color)
59     time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
60     bus.close() #close i2c bus 1
61
62 def StopScript():
63     bus.open(1) #open i2c bus 1
64     bus.write_byte(address, 0x6f, force=None) #stops the script
65     time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
66     bus.close() #close i2c bus 1
67
68
69 def LocationHelper():
70     bus.open(1) #open i2c bus 1
71     bus.write_i2c_block_data(address, 0x70, locator) #playing script locator
72     time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
73     bus.write_byte_data(address, 0x66, 30) #adjusting fade speed of the light script
74     time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
75     bus.write_byte_data(address, 0x74, 25) #adjusting time playback of the script
76     time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
77     bus.close() #close i2c bus 1
78
79 def FullHelper():
80     bus.open(1) #open i2c bus 1
81     bus.write_i2c_block_data(address, 0x70, full) #playing script locator
82     time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
83     bus.write_byte_data(address, 0x66, 30) #adjusting fade speed of the light script

```

```
84     time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
85     bus.write_byte_data(address, 0x74, 25) #adjusting time playback of the script
86     time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
87     bus.close() #close i2c bus 1
88
89     def StartSequence():
90         StopScript()
91         HSBFade(pink)
92         for item in AllColors:
93             SolidColor(item)
94
95     StopScript()
96     SolidColor(off)
97
```