```python
'''       __                          __
         /'{>                        /'{>
     ____) (____                 ____) (____
   //'--;   ;--'\\             //'--;   ;--'\\
  ///////\_/\\\\\\\           ///////\_/\\\\\\\
         m m                          m m

****************************************************************
*                                                              *
*  EEET-418                                                    *
*  27 April 2022                                               *
*  Senior Project: Smart package Delivery Box                  *
*                                                              *
*  The purpose of this program is to act as the BlinkM.py      *
*  file for the SPDB project. It is intended to be imported    *
*  into the main program.                                      *
*                                                              *
*  Created by: Brandon Empie                                   *
*                                                              *
****************************************************************
'''

import time
import smbus2
from smbus2 import SMBus, i2c_msg
address = 0x09
bus = smbus2.SMBus(1) #creating object


#Standard Colors
red = [0xff, 0x00, 0x00]
green = [0x00, 0xff, 0x00]
blue = [0x00, 0x00, 0xff]
white = [0xff, 0xff, 0xff]
cyan = [0x00, 0xff, 0xff]
yellow = [0xff, 0xff, 0x00]
purple = [0xff, 0x00, 0xff]
orange = [0xff, 0x12, 0x00]
off = [0x00, 0x00, 0x00]

AllColors = (red,white,blue,green,cyan,yellow,purple,orange,off)

#Special Circumstance Script
locater = [7, 0x00, 0x00]
full = [3, 0x00, 0x00]

#Fade to HSB Color
pink = [245, 0xff, 0xff]

def HSBFade(color):
    bus.open(1) #open i2c bus 1
    bus.write_i2c_block_data(address, 0x68, color)
    time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
    bus.close() #close i2c bus 1

def SolidColor(color):
    bus.open(1) #open i2c bus 1
    bus.write_i2c_block_data(address, 0x6e, color)
    time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
    bus.close() #close i2c bus 1

def StopScript():
    bus.open(1) #open i2c bus 1
    bus.write_byte(address, 0x6f, force=None) #stops the script
    time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
    bus.close() #close i2c bus 1


def LocationHelper():
    bus.open(1) #open i2c bus 1
    bus.write_i2c_block_data(address, 0x70, locater) #playing script locater
    time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
    bus.write_byte_data(address, 0x66, 30) #adjusting fade speed of the light script
    time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
    bus.write_byte_data(address, 0x74, 25) #adjusting time playback of the script
    time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
    bus.close() #close i2c bus 1

def FullHelper():
    bus.open(1) #open i2c bus 1
    bus.write_i2c_block_data(address, 0x70, full) #playing script locater
    time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
    bus.write_byte_data(address, 0x66, 30) #adjusting fade speed of the light script
    time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
    bus.write_byte_data(address, 0x74, 25) #adjusting time playback of the script
    time.sleep(.5) #waiting .5 seconds giving BlinkM time to process
    bus.close() #close i2c bus 1

def StartSequence():
    StopScript()
    HSBFade(pink)
    for item in AllColors:
        SolidColor(item)

StopScript()
SolidColor(off)
```