

```

#!/usr/bin/python
import tweepy , time, os, threading
from picamera import PiCamera
import RPi.GPIO as GPIO

# Setup GPIO LED and Motion PIR
LED = 26
PIR = 4
GPIO.setmode(GPIO.BCM)
GPIO.setup(PIR, GPIO.IN)
GPIO.setup(LED, GPIO.OUT)
GPIO.output(LED, GPIO.LOW)

# Set the list of users allowed to run commands
SuperUser = ['TWITTERID']

# Set the working directory for the script and files
WorkDir = "/FULL/PATH/WORKING/DIR"

# Blinking LED = 1 No Blinking LED = 0
BlinkLED = 1

def BlinkLed(OnOff, OffOn):
    # OnOff/OffOn will either blink on/off or off/on
    if BlinkLED == 1:
        GPIO.output(LED, OnOff)
        time.sleep(.5)
        GPIO.output(LED, OffOn)
        time.sleep(.5)

def TakePicture():
    # Say Cheese - we're gonna blink and then snap a pic
    PathToFile = WorkDir + "/mypic.JPG"
    for i in range(0,4):
        BlinkLed(0, 1) # Blink off then on
    camera.capture(PathToFile)
    GPIO.output(LED, 0) # led off

def GetStringID():
    # Need to know what the last tweet was that we used
    # so we don't reply to all previous tweets on startup
    # Open and read the file into the var NUMBER
    PathToFile = WorkDir + "/stringID.txt"
    try:
        file = open(PathToFile, "r")
        NUMBER = int(file.read())
        file.close()
    except IOError:
        NUMBER = 1 # If the file isn't there, start at 1
    return NUMBER

def SaveNewStringID(id_string):
    # Open and write over the old number with the new
    PathToFile = WorkDir + "/stringID.txt"
    file = open(PathToFile, "w")
    file.write(id_string)
    file.close()

def SendReply(user_name):
    # Send our tweet with an image
    PathToFile = WorkDir + "/mypic.JPG"
    tweet = "@" + user_name + " " + str(time.strftime("%c"))
    status = api.update_with_media(PathToFile, tweet)

def SendStatus(user_name):
    # Tweet current system status

```

```

if BlinkLED == 1:
    BLINK = "True"
else:
    BLINK = "False"
if motionDetection == 1:
    MD = "ON"
else:
    MD = "OFF"
tweet = "@" + user_name + " " + str(time.strftime("%c")) \
+ "\nFlip = " + str(camera.vflip) \
+ "\nBlink LED = " + BLINK \
+ "\nThread Running = " + ThreadRunning \
+ "\nMotion Detection = " + MD
status = api.update_status(status=tweet)

def UpdateUsers(newUser):
    # Add a new user to the Authorized user list and file
    PathToFile = WorkDir + "/Authorized.txt"
    Authorized.append(newUser)
    file = open(PathToFile, "w")
    file.write("")
    file.close
    file = open(PathToFile, "a")
    for u in Authorized:
        file.write(u + "\n")
    file.close()

def CamFlip():
    # This way we can flip the camera without changing the script
    if camera.vflip == True:
        camera.vflip = False
    else:
        camera.vflip = True

def CheckForCommands(commandString, user_name):
    global BlinkLED
    global motionDetection
    global stopThreads
    # Check the tweet to see if there were any commands we need to carry out
    tmpList = commandString.split(" ")
    cmdString = tmpList[0]
    cmdList = cmdString.split(":")
    if "cmd" in cmdList:
        if "adduser" in cmdList:           # Add a new twitter user ID to the Authorized
            UpdateUsers(cmdList[2])
            return 0
        elif "flip" in cmdList:           # Flip camera image vertically
            CamFlip()
            return 1
        elif "blinkoff" in cmdList:       # Disable LED blinking
            BlinkLED = 0
            return 0
        elif "blinkon" in cmdList:        # Enable LED blinking
            BlinkLED = 1
            return 0
        elif "alerton" in cmdList:         # Turn on the motion detector
            BlinkLED = 0
            motionDetection = 1
            SendStatus(user_name)
            return 0
        elif "alertoff" in cmdList:       # Turn off the motion detector
            BlinkLED = 1
            motionDetection = 0
            SendStatus(user_name)
            return 0
        elif "status" in cmdList:         # Send the current status

```

```

        SendStatus(user_name)
        return 0
    elif "reboot" in cmdList:        # Reboot the Pi
        GPIO.cleanup()
        os.system("reboot")
    elif "shutdown" in cmdList:    # Shutdown the Pi
        GPIO.cleanup()
        os.system("shutdown now")
    elif "stop" in cmdList:        # Stop the twitterPi.py script
        stopThreads = 1
        GPIO.cleanup()
        exit()
    else:
        return 0
else:
    return 1

def MonitorTweets():
    global id_string
    while True:
        search_text = "#YOURHASHTAG"
        search_result = api.search(search_text, rpp=1, since_id=id_string)
        for i in search_result:
            id_string = i.id_str
            SaveNewStringID(id_string)
            tweet = api.get_status(id_string)
            user_name = tweet.user.screen_name
            if user_name in Authorized:
                if user_name in SuperUser:
                    check = (CheckForCommands(i.text,user_name))
                else:
                    check = 1
                if check == 1:
                    TakePicture()
                    SendReply(user_name)
            BlinkLed(1,0) # Blink on then off
            time.sleep(30)
            BlinkLed(1, 0) # Who doesn't love a heartbeat?
            time.sleep(28)

def SecurityAlert():
    # This will run as a separate thread to monitor the
    # PIR (motion detector) when Security Alerts are on
    global ThreadRunning
    while True:
        try:
            if stopThreads == 1:
                exit()
            if motionDetection == 1:
                if GPIO.input(PIR):
                    time.sleep(1) # Want to wait a second before taking pic
                    TakePicture()
                    SendReply(SuperUser[0])
                    time.sleep(60)
        except:
            ThreadRunning = 'NO'
            SendStatus(SuperUser[0])
            exit()

# Set up the twitter API
CONSUMER_KEY = 'YOURCONSUMERKEY'
CONSUMER_SECRET = 'YOURCONSUMERSECRET'
ACCESS_KEY = 'YOURACCESSKEY'
ACCESS_SECRET = 'YOURACCESSSECRET'

auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)

```

```

auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)
api = tweepy.API(auth)

# Set up the PiCam
camera = PiCamera()
camera.vflip = True

# Read in the last tweet number we replied to
id_string = GetStringID()

# We only want authorized users to get replies
try:
    PathToFile = WorkDir + "/Authorized.txt"
    Authorized = []
    with open(PathToFile) as file:
        for line in file:
            line = line.strip()
            Authorized.append(line)
        file.close
except IOError:
    Authorized = [SuperUser[0]]

# Set the Motion Detection to Off
motionDetection = 0

# Set the ThreadRunning for the status requests.
ThreadRunning = "YES"

# Tweet that we are up and online.
SendStatus(SuperUser[0])

try:
    stopThreads = 0
    securityalert = threading.Thread(name='SecurityAlert', target=SecurityAlert)
    securityalert.start()
    MonitorTweets()
except KeyboardInterrupt:
    stopThreads = 1
    GPIO.cleanup() # Reset GPIO
    exit()

```