```
import RPi.GPIO as GPIO
import time
import serial
from mfrc522 import SimpleMFRC522
from gpiozero import MotionSensor
from picamera import PiCamera
from email.mime.multipart import MIMEMultipart
from subprocess import call
import os
import email.mime.application
import datetime
import smtplib
import I2C_LCD_driver

ser = serial.Serial("/dev/ttyAMA0",115200)
ser.flushInput()

phone_number = '' #********** change it to the phone number you want to call
text_message = ''
rec_buff = ''
power_key = 6

def send_at(command,back,timeout):
        rec_buff = ''
        ser.write((command+'\r\n').encode())
        time.sleep(timeout)

        if ser.inWaiting():
```

```
                time.sleep(0.01 )


                #print(rec_buff)
                rec_buff = ser.read(ser.inWaiting())
                #print(rec_buff)
        if rec_buff != '':
                print(rec_buff.decode())
                if 'red' in rec_buff.decode(): GPIO.output(23, GPIO.HIGH)
                if back not in rec_buff.decode():print(command + ' back:\t' +
rec_buff.decode())
                return 0
        else:
                #print(rec_buff)
                global TEXTDATA
                TEXTDATA = str(rec_buff)
                print(TEXTDATA)
                return 1


def ReceiveShortMessage():
        rec_buff = ''
        #print('Setting SMS mode...')
        send_at('AT+CMGF=1','OK',1)
        send_at('AT+CMGL="REC UNREAD"', 'OK', 1)
        answer = send_at('AT+CMGL="REC UNREAD"', '+CMTI', 1)



        if 1 == answer:
                answer = 0
```

```
                #print(rec_buff)
                if 'red' in rec_buff:
                        answer = 1
                        print('Turning LEDS onto RED')
        else:
                print('No New text')
                return False
        return True


def power_on(power_key):
        print('SIM7600X is starting:')
        GPIO.setmode(GPIO.BCM)
        GPIO.setwarnings(False)
        GPIO.setup(power_key,GPIO.OUT)
        time.sleep(0.1)
        GPIO.output(power_key,GPIO.HIGH)
        time.sleep(2)
        GPIO.output(power_key,GPIO.LOW)
        time.sleep(20)
        ser.flushInput()
        print('SIM7600X is ready')

def power_down(power_key):
        print('SIM7600X is loging off:')
        GPIO.output(power_key,GPIO.HIGH)
        time.sleep(3)
        GPIO.output(power_key,GPIO.LOW)
```

```
            time.sleep(18)
            print('Good bye')


def main():
    RELAY_PIN = 18
    piezo = 23

    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)

    MATRIX = [[1, 2, 3, 'A'],
          [4, 5, 6, 'B'],
          [7, 8, 9, 'C'],
          ['*', 0, '#', 'D']]

    COL = [26, 19, 13, 5]
    ROW = [21, 20, 16, 12]

    GPIO.setup(RELAY_PIN, GPIO.OUT)
    GPIO.setup(piezo, GPIO.OUT)

    # Secret Code
    secret_code = [1, 'A', 2, 'B']  # Secret code is 12AB

    input_buffer = []

    a = 0  # iterations
```

```
# Create objects for RFID, motion sensor, LCD, and camera
read = SimpleMFRC522()
pir = MotionSensor(17)
camera = PiCamera()
lcd = I2C_LCD_driver.lcd()

from_email_addr = 'raspberrypi031402@gmail.com'
from_email_password = 'wbdp hlhr qpwz dpmq'
to_email_addr = 'songzs758@gmail.com'

Tag_ID = "1046809351663"
door = False
lockdown_initiated = False

while True:
    lcd.lcd_clear()
    lcd.lcd_display_string("Place your Tag", 1, 1)
    id, Tag = read.read()

    id = str(id)

    if id == Tag_ID:
        lcd.lcd_clear()
        lcd.lcd_display_string("Successful", 1, 3)

        if door:
            lcd.lcd_display_string("Door is locked", 2, 1)
```

```
    time.sleep(0.5)
    door = False
    time.sleep(3)
else:
    lcd.lcd_display_string("Door is open", 2, 2)
    time.sleep(0.5)
    door = True
    time.sleep(3)

if pir.motion_detected:
    print("Motion Detected")

    camera.resolution = (640, 480)
    camera.rotation = 180
    camera.start_recording('alert_video.h264')
    camera.wait_recording(5)
    camera.stop_recording()

    command = "MP4Box -add alert_video.h264 alert_video.mp4"
    call(command, shell=True)
    print("video converted")

    msg = MIMEMultipart()
    msg['Subject'] = 'INTRUDER ALERT..!!'
    msg['From'] = from_email_addr
    msg['To'] = to_email_addr

    Captured = '/home/raspberrypi/Desktop/alert_video.mp4'
```

```
        fp = open(Captured, 'rb')
        att = email.mime.application.MIMEApplication(fp.read(), _subtype=".mp4")
        fp.close()
        att.add_header('Content-Disposition', 'attachment', filename='video' +
datetime.datetime.now().strftime(
            '%Y-%m-%d%H:%M:%S') + '.mp4')
        msg.attach(att)

        print("attach successful")

        os.remove("/home/raspberrypi/Desktop/alert_video.h264")

        os.rename('alert_video.mp4',
              datetime.datetime.now().strftime('%Y-%m-%d%H:%M:%S') + '.mp4')

        server = smtplib.SMTP('smtp.gmail.com', 587)
        server.starttls()
        server.login(from_email_addr, from_email_password)
        server.sendmail(from_email_addr, to_email_addr, msg.as_string())
        server.quit()
        print('Email sent')

        timeout1 = 10
        start_time = time.time()

        while time.time() - start_time < timeout1:
            ReceiveShortMessage()
```

```
        # Set up keypad
        for j in range(4):
            GPIO.setup(COL[j], GPIO.OUT)
            GPIO.output(COL[j], 1)

        for i in range(4):
            GPIO.setup(ROW[i], GPIO.IN, pull_up_down=GPIO.PUD_UP)

        try:
            # Introduce a buffer time for entering keypad password
            start_time = time.time()
            timeout = 60  # 60 seconds (1 minute)
            code_entered = False

            while not lockdown_initiated and time.time() - start_time < timeout and not
code_entered:
                for j in range(4):
                    GPIO.output(COL[j], 0)

                    for i in range(4):
                        if GPIO.input(ROW[i]) == 0:
                            input_buffer.append(MATRIX[i][j])
                            print(MATRIX[i][j])
                            while GPIO.input(ROW[i]) == 0:
                                pass
                            time.sleep(0.3)

                    GPIO.output(COL[j], 1)
```

```
            if len(input_buffer) >= len(secret_code):
                if input_buffer[-len(secret_code):] == secret_code:
                    print("Secret code matched!")
                    code_entered = True
                    break
                else:
                    if a == 2:
                        lockdown_initiated = True
                        print("Lockdown Initiated!")
                        GPIO.output(RELAY_PIN, 1)
                        GPIO.output(piezo, 1)
                        time.sleep(2)

                        ReceiveShortMessage()



                    else:
                        a += 1
                        print("Secret code does not match!")
                        input_buffer = []

            if time.time() - start_time >= timeout:
                print("Timeout: No code entered within the allotted time.")

    except KeyboardInterrupt:
        GPIO.cleanup()
```

```
        else:
            lcd.lcd_clear()
            lcd.lcd_display_string("Wrong Tag!", 1, 3)
            time.sleep(0.3)

if __name__ == "__main__":
    power_on(power_key)
    main()
```