Raspberry Pi project coding.

```
import telepot
from picamera import PiCamera
import RPi.GPIO as GPIO
import time
from time import sleep
from telepot.loop import MessageLoop
from subprocess import call
import threading
import smtplib
from email.message import EmailMessage
import os

# Email setup
EMAIL_ADDRESS = 'agunwaxammy72@gmail.com'
EMAIL_PASSWORD = 'dedp cttj tcss ldts'
EMAIL_RECEIVER = 'recipient@example.com'

def send_email(subject, body, attachment_path=None):
    msg = EmailMessage()
    msg['Subject'] = subject
    msg['From'] = EMAIL_ADDRESS
    msg['To'] = EMAIL_RECEIVER
    msg.set_content(body)

    if attachment_path and os.path.exists(attachment_path):
        with open(attachment_path, 'rb') as f:
            file_data = f.read()
            file_name = os.path.basename(attachment_path)
            # Use correct type based on extension
            if file_name.endswith(".jpg"):
                maintype, subtype = 'image', 'jpeg'
            else:
                maintype, subtype = 'application', 'octet-stream'
            msg.add_attachment(file_data, maintype=maintype, subtype=subtype,
filename=file_name)

    try:
        with smtplib.SMTP_SSL('smtp.gmail.com', 465) as smtp:
            smtp.login(EMAIL_ADDRESS, EMAIL_PASSWORD)
            smtp.send_message(msg)
            print("📨 Email sent successfully.")
    except Exception as e:
        print("❌ Failed to send email:", e)

# GPIO and Camera setup
IR = 4
BUZZER = 6

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

GPIO.setup(IR, GPIO.IN)
GPIO.setup(BUZZER, GPIO.OUT)
GPIO.output(BUZZER, GPIO.LOW)

motion = 1
motionNew = 1
alarm_enabled = False

# Initialize camera,
camera = PiCamera()

bot = telepot.Bot('7584611319:AAHAsG3dEUw5UH8Ad2itNFcKPps0602587A')

def sendNotification():
    global camera  # Ensure the camera is used correctly within the thread
    timestamp = time.strftime("%y%b%d_%H%M%S")

    # Capture snapshot
    photo_file = f"./snapshot_{timestamp}.jpg"
    camera.capture(photo_file)
    print(f"Photo taken: {photo_file}")

    # Send photo on Telegram
    bot.sendPhoto(chat_id, photo=open(photo_file, 'rb'))
    bot.sendMessage(chat_id, '🚨 An Intruder detected at faculty of engineering security cam')

    # Send photo by email
    send_email(
        subject="Security Alert: Motion Detected",
        body="An Intruder detected at faculty of engineering security cam.",
        attachment_path=photo_file
    )

def monitor_motion():
    global motion, motionNew, chat_id, alarm_enabled
    while True:
        if GPIO.input(IR) == 0 and alarm_enabled:
            print("Motion detected")
            motion = 0
            if motionNew != motion:
                motionNew = motion
                sendNotification()
        else:
            print("✅ No motion")
            motion = 1
            if motionNew != motion:
                motionNew = motion
        sleep(1)  # To avoid 100% CPU usage and to give time for GPIO to change

def handle(msg):
    global chat_id, alarm_enabled
    chat_id = msg['chat']['id']
    text = msg['text'].lower()

    print(f"📩 Message received: {text} from {chat_id}")

    if text == '/start':
        bot.sendMessage(chat_id, '📷 Security system is active.\nUse /activate or /alarm_on to
activate.')
    elif text == '/activate':
        alarm_enabled = True
        bot.sendMessage(chat_id, '🔔 Motion detection enabled.')
    elif text == '/deactivate':
        alarm_enabled = False
        bot.sendMessage(chat_id, '🔕 Motion detection disabled.')
    elif text == '/alarm_on':
        GPIO.output(BUZZER, GPIO.HIGH)
        sleep(0.5) # Delay in seconds
        GPIO.output(BUZZER,GPIO.LOW)
        sleep(0.5)
        #bot.sendMessage(chat_id, '📢 Buzzer turned ON.')
    elif text == '/alarm_off':
        GPIO.output(BUZZER, GPIO.LOW)
        bot.sendMessage(chat_id, '🔇 Buzzer turned OFF.')
    elif text == '/status':
        buzzer_state = GPIO.input(BUZZER)
        status_msg = f"🔔 Alarm: {'ON' if alarm_enabled else 'OFF'}\n📢 Buzzer: {'ON' if
buzzer_state else 'OFF'}"
        bot.sendMessage(chat_id, status_msg)
    else:
bot.sendMessage(chat_id, '❓ Unknown command.\nAvailable: /start, /activate, /deactivate, /alarm_on, /alarm_off, /status')

# Start bot and motion detection thread
MessageLoop(bot, handle).run_as_thread()
print('🤖 Bot is listening...')
motion_thread = threading.Thread(target=monitor_motion)
motion_thread.start()

# Keep script running
while True:
    time.sleep(10)
```