

Smart Intercom System - Python Code

```
#!/usr/bin/env python3
import serial
import time
from picamera import PiCamera
import os
from telegram import Bot
from telegram.ext import Updater, CommandHandler

last_time_button_pressed = time.time()
button_pressed_delay = 5.0 #seconds

open_door_request = False
handling_door = False

# Init Serial
print("Trying to connect to Serial.")
while True:
    try:
        ser = serial.Serial('/dev/ttyACM0', 115200, timeout=1.0)
        print("Successfully connected to Serial.")
        break
    except serial.SerialException:
        print("Could not connect to Serial. Retrying in 1 second...")
        time.sleep(1)

def send_to_arduino(text):
    str_to_send = text.rstrip() + "\n"
    ser.write(str_to_send.encode('utf-8'))

# Init Camera
print("Init camera.")
camera = PiCamera()
camera.resolution = (640, 480)
camera.rotation = 180
image_folder_name = "/home/pi/camera"
if not os.path.exists(image_folder_name):
    os.mkdir(image_folder_name)
image_file_name = image_folder_name + "/door_camera.jpg"
print("Camera OK.")

# Telegram callbacks
def start_handler(update, context):
    str_to_send = "Welcome " + update.effective_user.first_name + "!"
    context.bot.send_message(chat_id=update.effective_chat.id, text=str_to_send)

def open_door_handler(update, context):
    global open_door_request
    global handling_door
    if open_door_request and not handling_door:
        print("Opening the door.")
        handling_door = True
        send_to_arduino("open_door")
        send_to_arduino("print_text:Door is opened.")
        send_to_arduino("play_buzzer:300,500")
        send_to_arduino("set_led:000,255,000")
        bot.send_message(chat_id=update.effective_chat.id, text="Opening the door.")
        time.sleep(10)
        print("Closing the door.")
        send_to_arduino("close_door")
        send_to_arduino("print_text:Push on button to call.")
        send_to_arduino("play_buzzer:200,500")
        send_to_arduino("set_led:000,000,255")
        open_door_request = False
        handling_door = False
```

```

def deny_access_handler(update, context):
    global open_door_request
    global handling_door
    if open_door_request and not handling_door:
        print("Denying access.")
        handling_door = True
        send_to_arduino("print_text:Access denied.")
        send_to_arduino("play_buzzer:200,1000")
        send_to_arduino("set_led:255,000,000")
        bot.send_message(chat_id=update.effective_chat.id, text="Denying access.")
        time.sleep(5)
        send_to_arduino("print_text:Push on button to call.")
        send_to_arduino("set_led:000,000,255")
        open_door_request = False
        handling_door = False

# Init Telegram Bot
print("Init Telegram bot.")
chat_id = "-623214910"
with open("/home/pi/.local/share/.telegram_bot_token", "r") as f:
    telegram_token = f.read().rstrip()
bot = Bot(token=telegram_token)
updater = Updater(token=telegram_token)
dispatcher = updater.dispatcher
dispatcher.add_handler(CommandHandler('start', start_handler))
dispatcher.add_handler(CommandHandler('open', open_door_handler, run_async=True))
dispatcher.add_handler(CommandHandler('deny', deny_access_handler, run_async=True))

# Wait after setup
print("Waiting for 3 seconds...")
time.sleep(3)
ser.reset_input_buffer()
updater.start_polling()
send_to_arduino("print_text:Push on button to call.")
send_to_arduino("set_led:000,000,255")
print("OK. Main loop is starting.")

# Run main loop
try:
    while True:
        time.sleep(0.01)
        if ser.in_waiting > 0:
            msg = ser.readline().decode('utf-8').rstrip()
            if msg == "button_pressed":
                time_now = time.time()
                if (time_now - last_time_button_pressed >= button_pressed_delay) and \
                   (not open_door_request) and (not handling_door):
                    last_time_button_pressed = time_now
                    print("Open door request. Taking a photo and sending it to Telegram.")
                    open_door_request = True
                    send_to_arduino("print_text:Please wait for a few seconds.")
                    camera.capture(image_file_name)
                    bot.send_message(chat_id=chat_id, text="Someone is ringing the bell!")
                    with open(image_file_name, 'rb') as photo:
                        bot.send_photo(chat_id=chat_id, photo=photo)
                    bot.send_message(chat_id=chat_id, text="Send /open or /deny")
            else:
                ser.reset_input_buffer()
except KeyboardInterrupt:
    print("---")
    print("Closing Serial communication.")
    ser.close()
    print("Stopping Telegram updater, wait a few seconds...")
    updater.stop()
    print("End of program.")

```