

Отчет по практической работе №3

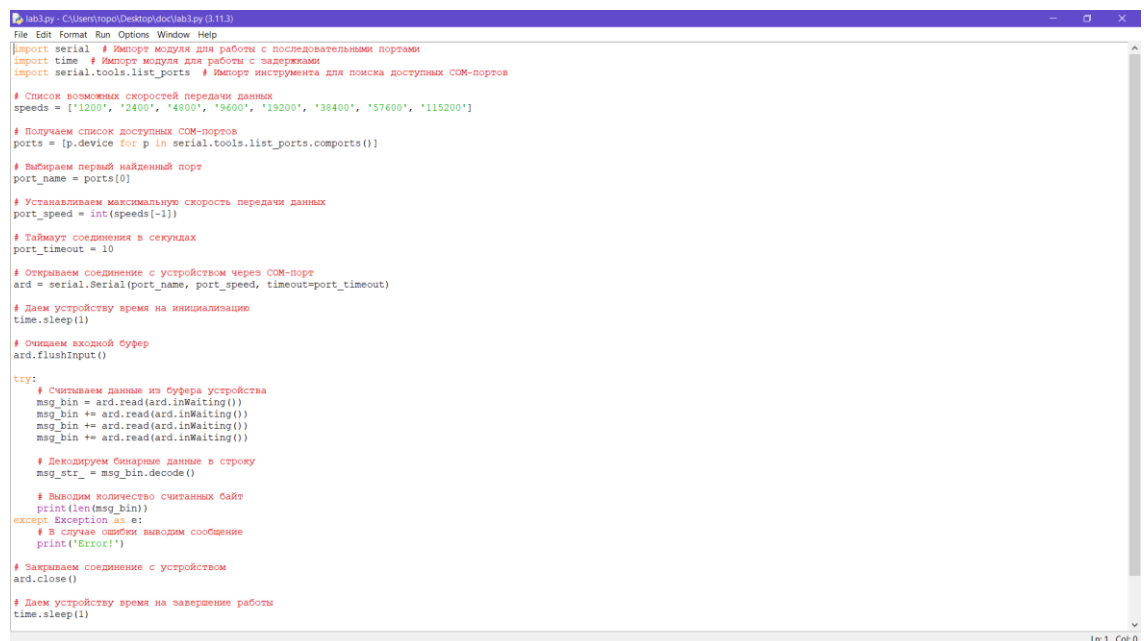
Тема: Создание и тестирование многострочной Python-программы, контейнеризация в Docker, работа с Anaconda и Jupyter Notebook

Цель работы:

1. Создать и протестировать программу на Python для работы с последовательными портами.
2. Упаковать программу в Docker-контейнер.
3. Разместить проект в GitHub.
4. Изучить основы работы с Anaconda и Jupyter Notebook.

Ход работы

1. Создание многострочной программы на Python



```
lab3.py - C:\Users\ropo\Desktop\doc\lab3.py (3.11.3)
File Edit Format Run Options Window Help

import serial # Импорт модуля для работы с последовательными портами
import time # Импорт модуля для работы с задержками
import serial.tools.list_ports # Импорт инструмента для поиска доступных COM-портов

# Список возможных скоростей передачи данных
speeds = ['1200', '2400', '4800', '9600', '19200', '38400', '57600', '115200']

# Получаем список доступных COM-портов
ports = [p.device for p in serial.tools.list_ports.comports()]

# Выбираем первый найденный порт
port_name = ports[0]

# Устанавливаем максимальную скорость передачи данных
port_speed = int(speeds[-1])

# Таймаут соединения в секундах
port_timeout = 10

# Открываем соединение с устройством через COM-порт
ard = serial.Serial(port_name, port_speed, timeout=port_timeout)

# Даем устройству время на инициализацию
time.sleep(1)

# Очищаем входной буфер
ard.flushInput()

try:
    # Считываем данные из буфера устройства
    msg_bin = ard.read(ard.inWaiting())
    msg_bin += ard.read(ard.inWaiting())
    msg_bin += ard.read(ard.inWaiting())
    msg_bin += ard.read(ard.inWaiting())

    # Декодируем бинарные данные в строку
    msg_str = msg_bin.decode()

    # Выводим количество считанных байт
    print(len(msg_bin))
except Exception as e:
    # В случае ошибки выводим сообщение
    print('Error!')

# Закрываем соединение с устройством
ard.close()

# Даем устройству время на завершение работы
time.sleep(1)
```

Рис. 1 – Создание программы на Python

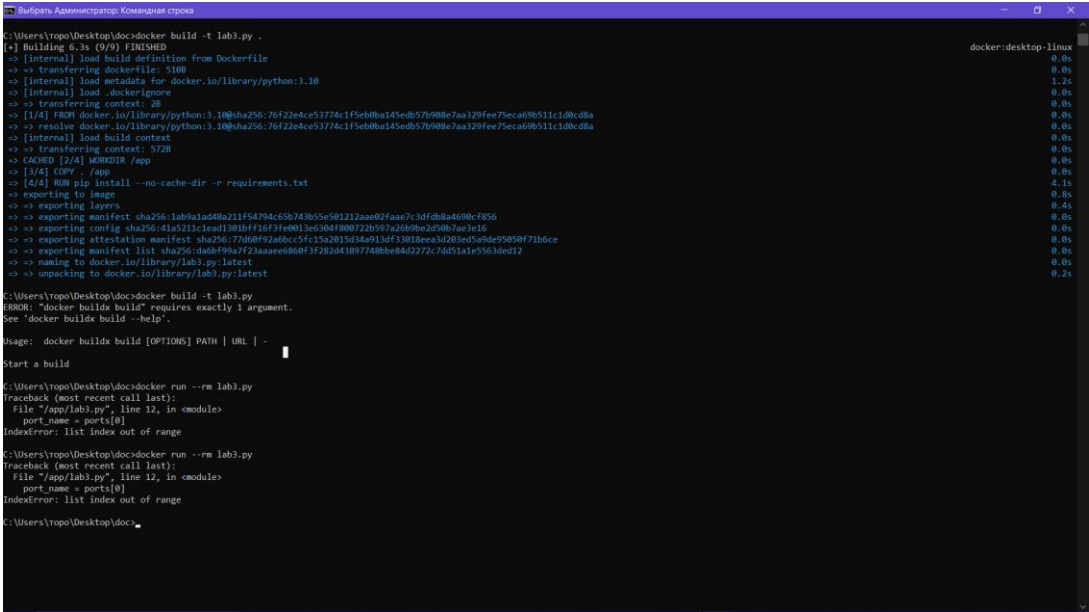
2. Запуск программы

Программа была запущена, но так как на устройстве не было открытых COM-портов, работа была прекращена с ошибкой:

IndexError: list index out of range

Это значит, что `serial.tools.list_ports.comports()` не обнаружил доступных портов.

3. Упаковка программы в Docker-контейнер



```
C:\Users\topo\Desktop>docker build -t lab3.py .
[+] Building 6.3s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/python:3.10
=> [internal] load metadata for docker.io/library/python:3.10
=> [internal] load build context
=> [1/4] FROM docker.io/library/python:3.10@sha256:76f2264c53774c1f5eb0ba145ed57b908e7aa1294ee75ec4606511c1d0c8ba
=> resolve docker.io/library/python:3.10@sha256:76f2264c53774c1f5eb0ba145ed57b908e7aa1294ee75ec4606511c1d0c8ba
=> [internal] load build context
=> transferring context: 572B
=> CACHED [2/4] FROM python:3.10
=> [3/4] COPY . /app
=> [4/4] RUN pip install --no-cache-dir -r requirements.txt
=> exporting to image
=> exporting layers
=> exporting manifest sha256:1ab9a1ad48a211f5479dc65b743b55e501212aae02faa7c3dfdb8a460c4f856
=> exporting config sha256:41a5211c4e011b1b1f1f1d1f1e013e5b8f080720197a2d08e2650b7e0e1e16
=> exporting attestation manifest sha256:77d00f92abcc5f15a2015d5a9134f33018ee341201ed5a4e95090f71bdce
=> exporting manifest list sha256:dab0f99a7f23aaee6800f3f282043097740bbe84d227c7dd51a1e5561ded12
=> naming to docker.io/library/lab3.py:latest
=> unpacking to docker.io/library/lab3.py:latest

C:\Users\topo\Desktop>docker build -t lab3.py
ERROR: "docker buildx build" requires exactly 1 argument.
See "docker buildx build --help".

Usage: docker buildx build [OPTIONS] PATH | URL | -

Start a build

C:\Users\topo\Desktop>docker run --rm lab3.py
Traceback (most recent call last):
  File "/app/lab3.py", line 12, in <module>
    port_name = ports[0]
IndexError: list index out of range

C:\Users\topo\Desktop>docker run --rm lab3.py
Traceback (most recent call last):
  File "/app/lab3.py", line 12, in <module>
    port_name = ports[0]
IndexError: list index out of range

C:\Users\topo\Desktop>
```

Рис. 2 – Упаковка программы в Docker-контейнер

4. Размещение проекта в GitHub

Разместили папку с 3 практической работой в нашем первом репозитории, который был создан для предмета Интеллектуальные системы и технологии

5. Основы Anaconda и Jupyter Notebook

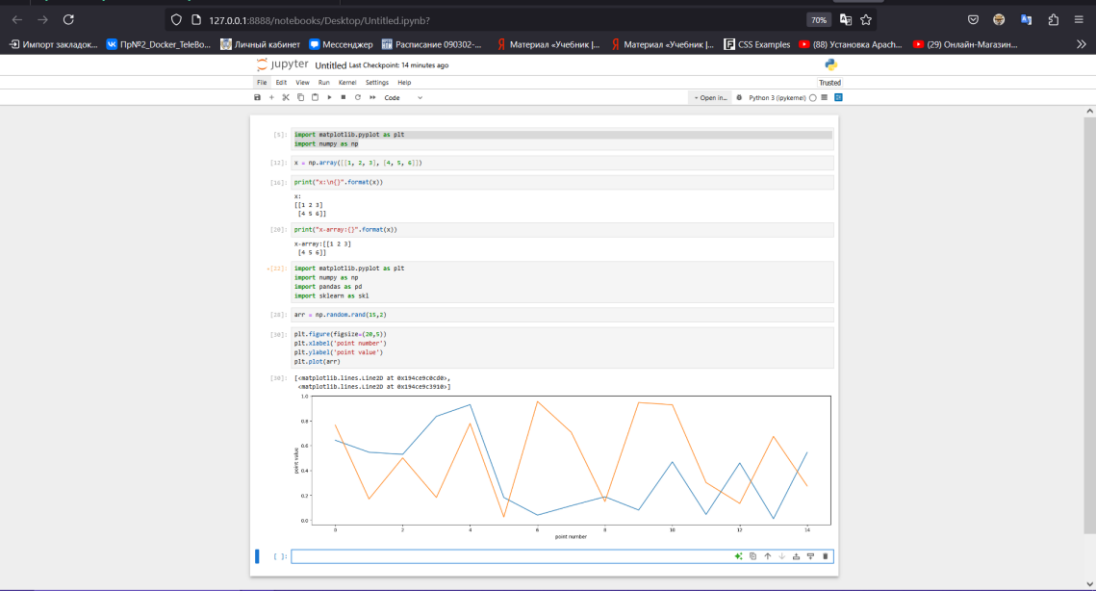


Рис. 3 – Построение графика в Jupyter Notebook.

Вывод

В ходе работы была создана и протестирована Python-программа, упакована в Docker-контейнер и размещена в репозитории GitHub. Также были изучены основы работы с Anaconda и Jupyter Notebook.