



Федеральное государственное автономное образовательное учреждение высшего образования «Национальный Исследовательский Университет ИТМО»

**ЛАБОРАТОРНАЯ РАБОТА №4**  
**«ПРОГРАММИРОВАНИЕ ПРОМЫШЛЕННЫХ РОБОТОВ»**  
**«СЕТЕВЫЕ СООБЩЕНИЯ»**

Преподаватель:  
Громов В. С.

Выполнили:  
Румянцев А. А., R3341  
Чебаненко Д. А., R3341  
Овчинников П. А., R3341  
Блохин С. О., R3342  
Тоскано О. Арасели Д. П., R3338

Факультет: СУиР

Санкт-Петербург  
2024

## Содержание

<b>1 Цели выполнения работы</b>	<b>2</b>
<b>2 Код конечной программы</b>	<b>2</b>
2.1 MELFA BASIC . . . . .	2
2.2 Python . . . . .	3
2.3 Описание команд . . . . .	4
<b>3 Таблица сохраненных точек</b>	<b>4</b>
<b>4 Этапы выполнения программы</b>	<b>5</b>
<b>5 Выводы</b>	<b>9</b>

# 1 Цели выполнения работы

Написать программу, которая будет считывать из TCP/IP порта 10003 номера деталей по осям X и Y (от 0 до 2 по каждой из осей), затем переносить указанную деталь с первого стола с деталями на второй пустой стол. На втором столе детали следует выкладывать последовательно линиями. Указанные операции должны производиться 9 раз (по общему числу деталей).

## 2 Код конечной программы

### 2.1 MELFA BASIC

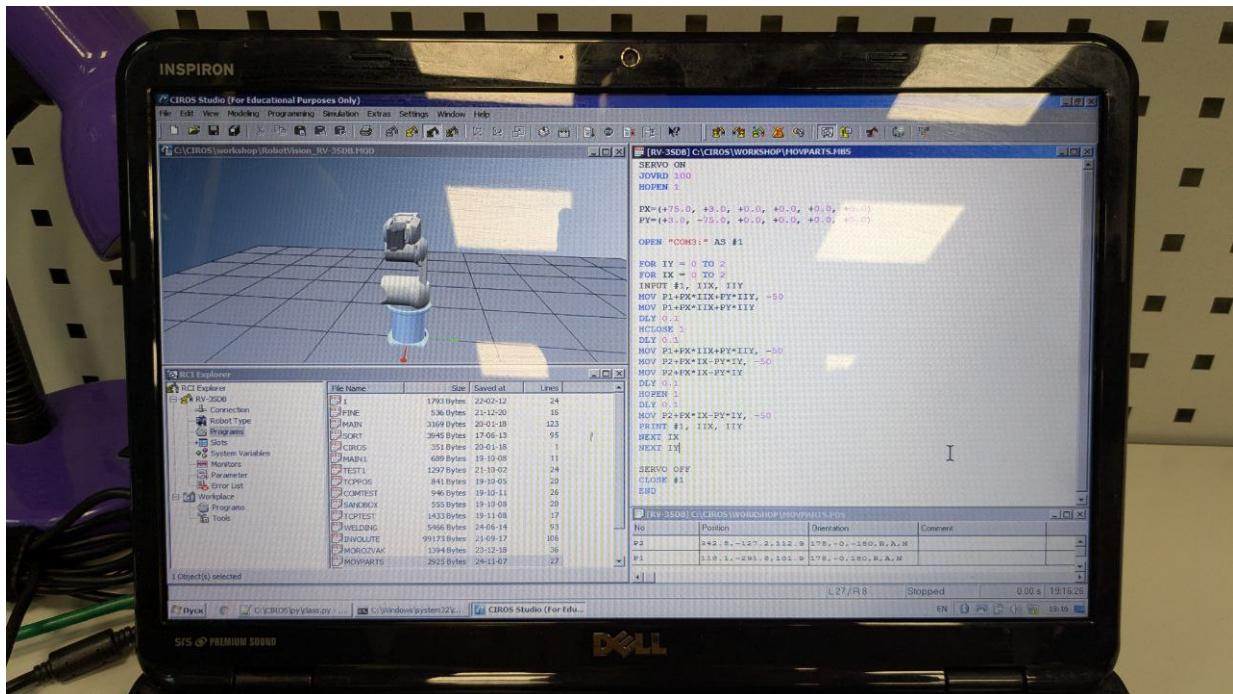


Рис. 1: Фото конечной программы на языке MELFA BASIC

```

SERVO ON
JOVRD 100
HOPEN 1

PX=(+75.0, +3.0, +0.0, +0.0, +0.0, +0.0)
PY=(+3.0, -75.0, +0.0, +0.0, +0.0, +0.0)

OPEN "COM3:" AS #1

FOR IY = 0 TO 2
FOR IX = 0 TO 2
INPUT #1, IIX, IIY
MOV P1+PX*IIX+PY*IIY, -50
MOV P1+PX*IIX+PY*IIY
DLY 0.1
HCLOSE 1
DLY 0.1
MOV P1+PX*IIX+PY*IIY, -50
PRINT #1, IIX, IIY
NEXT IX
NEXT IY

SERVO OFF
CLOSE #1
END

```

```

MOV P2+PX*IX-PY*IY
DLY 0.1
HOPEN 1
DLY 0.1
MOV P2+PX*IX-PY*IY, -50
PRINT #1, IIX, IIY
NEXT IX
NEXT IY

SERVO OFF
CLOSE #1
END

```

Листинг 1: Листинг конечной программы на языке MELFA BASIC

## 2.2 Python

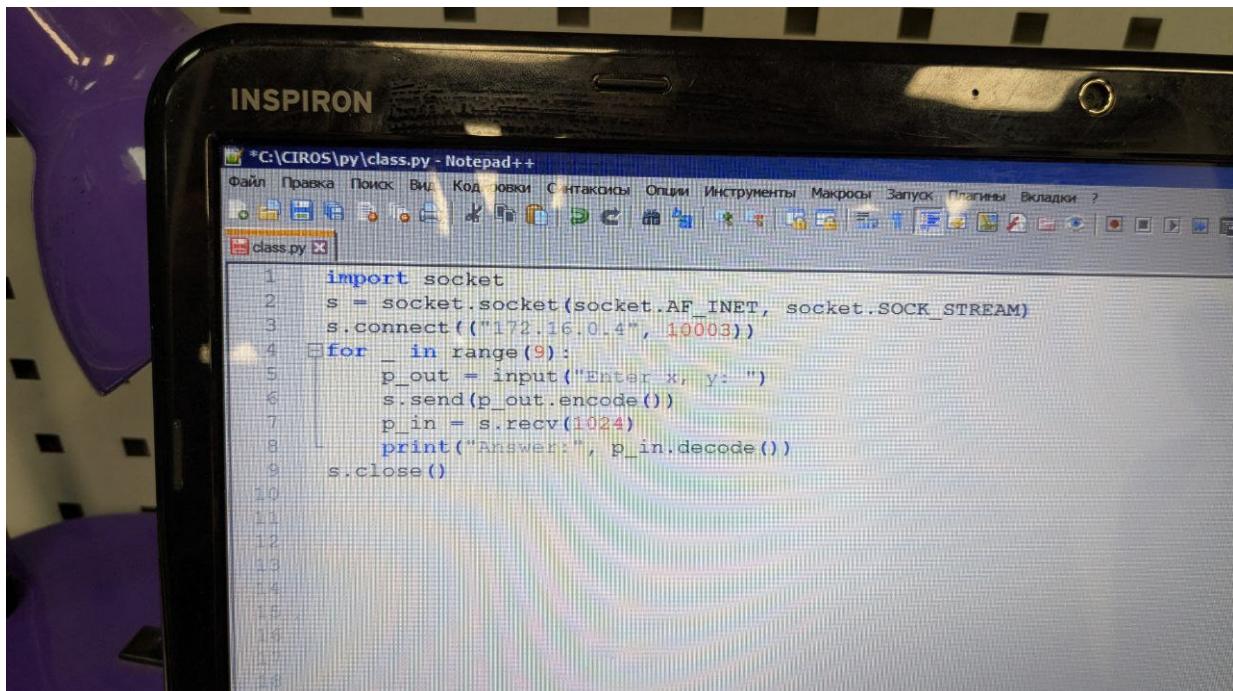


Рис. 2: Фото конечной программы на языке Python

```

import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("172.16.0.4", 10003))
for _ in range(9):
    p_out = input("Enter x, y: ")
    s.send(p_out.encode())
    p_in = s.recv(1024)
    print("Answer:", p_in.decode())
s.close()

```

Листинг 2: Листинг конечной программы на языке Python

## 2.3 Описание команд

При выполнении лабораторной работы использовались следующие команды программирования языка MELFA BASIC:

- END – завершение программы
- SERVO ON – включение двигателей
- JOVRD 100 – скорость движения в процентах от максимальной
- SERVO OFF – выключение двигателей
- DLY 0.1 – пауза выполнения программы в секундах
- HOPEN 1 – открытие захватного устройства
- HCLOSE 1 – закрытие захватного устройства
- PX=(+75.0, +3.0, +0.0, +0.0, +0.0, +0.0) – вспомогательная переменная координат формата (X, Y, Z, A, B, C) с декартовыми координатами для создания настраиваемого смещения координат
- OPEN "COM3:"AS #1 – открытие TCP/IP порта 10003 для подключения интерфейса #1
- CLOSE #1 – закрытие TCP/IP порта 10003 для подключения интерфейса #1
- FOR IY = 0 TO 2 – начало выполнения цикла, IY – переменная итерации цикла
- NEXT IY – окончание цикла
- INPUT #1, IIX, IIY – считывание содержимого буфера интерфейса #1 в переменные IIX и IIY
- PRINT #1, IIX, IIY – вывод содержимого переменных IIX и IIY в интерфейс #1
- MOV P1+PX\*IIX+PY\*IIY – движение в точку P1 из таблицы сохраненных точек с положительным смещением PX\*IIX+PY\*IIY
- MOV P1+PX\*IIX+PY\*IIY, -50 – движение в точку P1 из таблицы сохраненных точек с положительным смещением PX\*IIX+PY\*IIY и 50 мм вверх по оси Z

А также следующие команды программирования языка Python:

- import socket – импорт модуля для работы с сетевыми соединениями
- s = socket.socket(...) – создаем TCP-сокет для соединения с роботом
- s.connect(...) – подключаемся к роботу по IP адресу и порту
- for \_ in range(9) – цикл с количеством итераций равным количеству деталей
- p\_out = input(...) – запрос у пользователя ввода координат
- s.send(p\_out.encode()) – отправка роботу строки в виде байтового формата
- p\_in = s.recv(1024) – получение ответа от робота (макс. размер данных 1024 байта)
- print(...) – вывод ответа робота в консоль
- s.close() – закрытие соединения с роботом

## 3 Таблица сохраненных точек

No	Position	Orientation	Comment
P2	342.6, -127.2, 112.9	178, -0, -180, R, A, N	
P1	118.1, -291.8, 101.9	178, -0, 180, R, A, N	

Рис. 3: Фото таблицы с сохраненными точками

No	Position	Orientation
P1	118.1, -291.8, 101.9	178, -0, 180, R, A, N
P2	342.6, -127.2, 112.9	178, -0, -180, R, A, N

## 4 Этапы выполнения программы

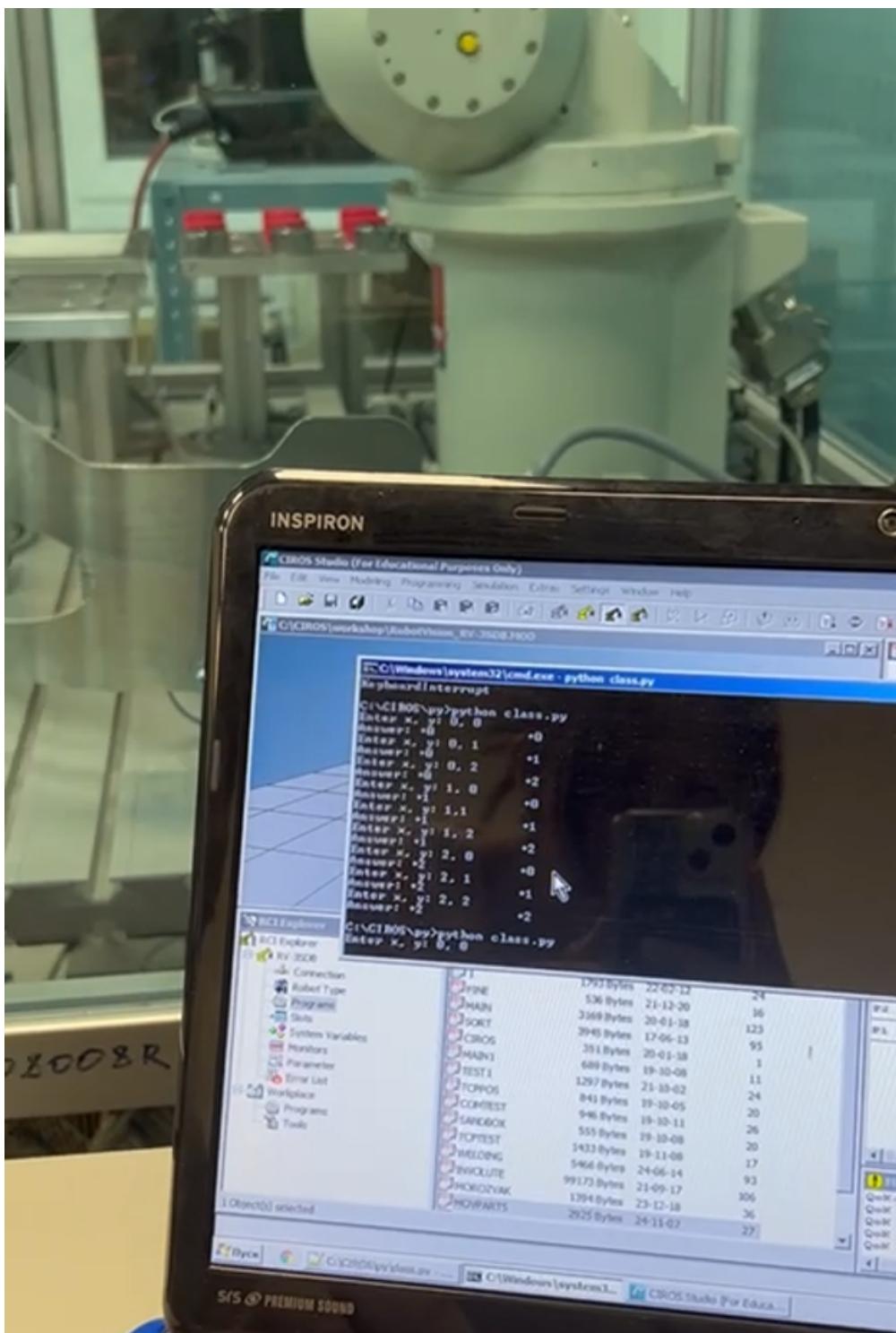


Рис. 4: Ввод координаты точки (0, 0) в консоль с запущенным Python скриптом

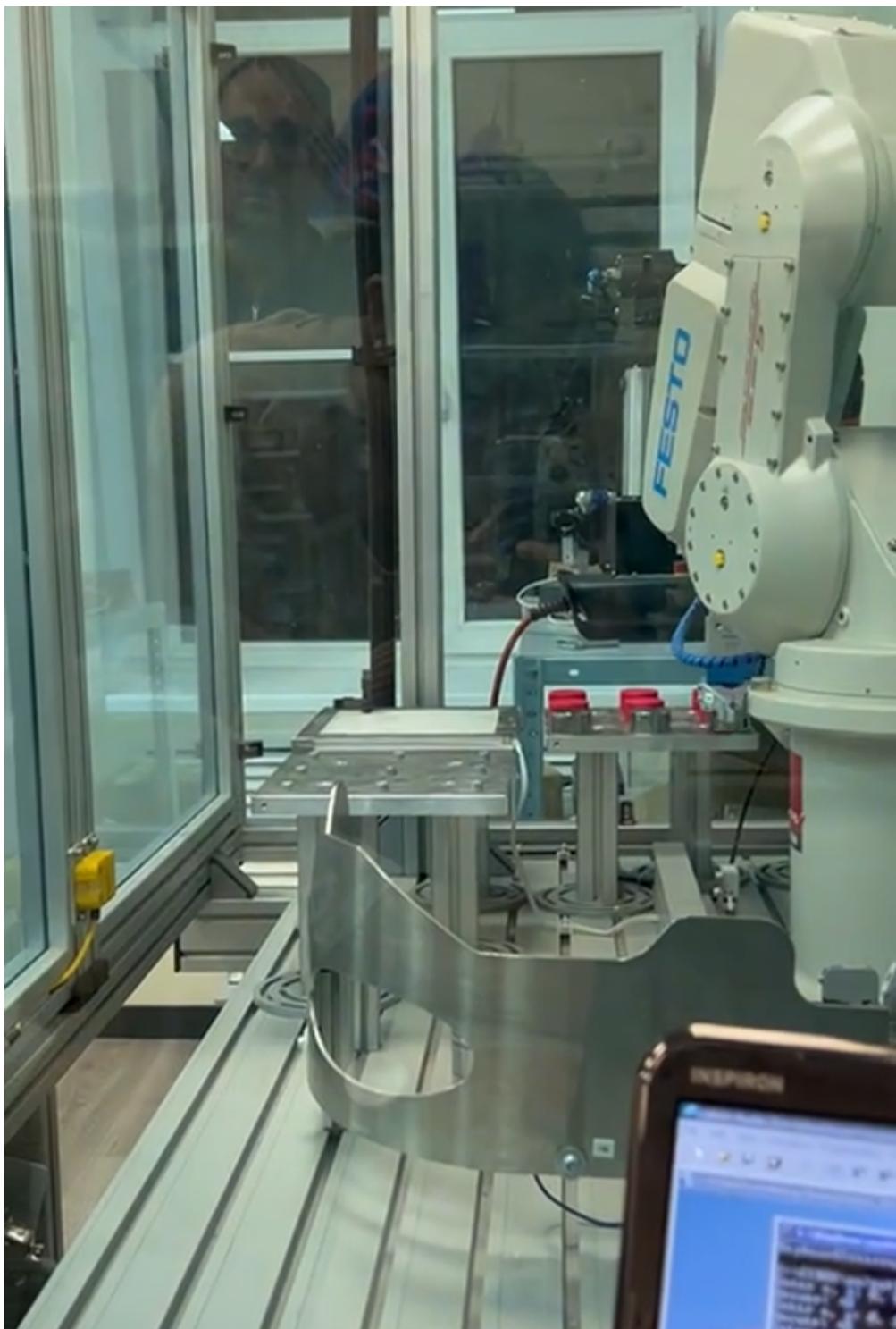


Рис. 5: Робот берет деталь с позиции (0, 0)

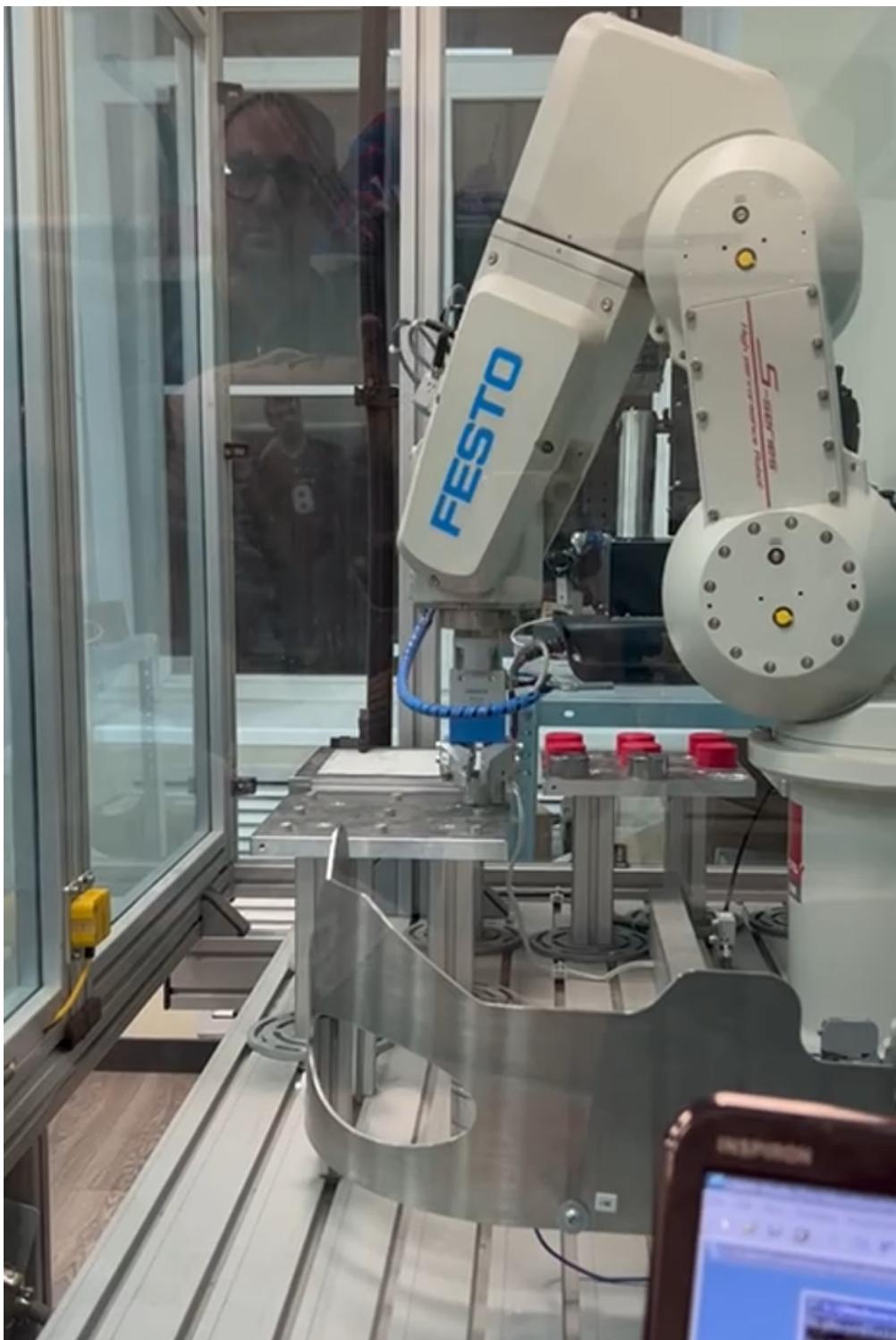


Рис. 6: Робот кладет деталь в начальную для последовательного переноса точку



Рис. 7: Робот продолжает переносить детали с задаваемых координат с первого стола на второй, размещая их последовательно

## 5 Выводы

В результате выполнения лабораторной работы мы:

- познакомились с передачей TCP/IP пакетов с компьютера на робот и обратно
- написали программу на языке MELFA BASIC для размещения роботом деталей на второй стол в последовательном порядке в соответствии с вводимыми в консоль с запущенным скриптом на языке Python координатами деталей с первого стола