

Project Raspberry Pi

Instructor: Prof. Dr Sarosh Patel

Aalekh Mishra

1173657

Raspberry Pi Serial Port Connection

Objective: To send a function using serial port of RS232 cable through Raspberry Pi. We must emulate the sent function on a note pad or a keyboard. Use an HMI feature to enhance the project.

Credentials:

Account: aamishra

Password: 122627

Tools Required:

- Raspberry Pi 3
- Arduino
- Jump wires
- Button (if needed)
- RS232 Cable
- Raspberry Pi Screen or Monitor
- Power Supply for Raspberry Pi

(Until you connect Raspberry Pi and laptop which has VS code to the same network, you will be needing a monitor. Once they are connected on the same network, the monitor is not a necessity).

Software needed:

- Real VNC Viewer
- Putty
- Arduino IDE
- VS code
- Thonny

Connection and Pin Out: At the raspberry end, we should be using the USB port as the Pi doesn't have a serial port. To attach a serial port, you need an RS232 cable. Whose RX and TX pin will be connected to Arduino at the RX and TX pin respectively, doing vice versa will not generate any form of output at the laptop. The Arduino will be connected using a USB cable to the computer where the Arduino IDE will be recognized by the computer. Since it's a serial Port connection Pins of Raspberry Pi will not be used. After all the connections are made, power up the Pi. Use a USB type keyboard and a USB type mouse to navigate through its desktop.

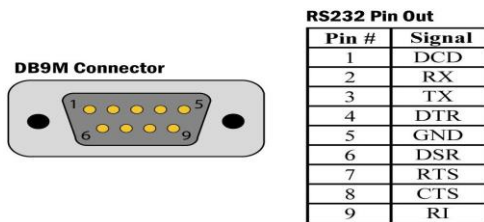
Raspberry Pi:

After powering up the Pi use the credentials above to log in and run prog5.py on the desktop. It will have arrow keys and a text box to enter text. Use Thonny to run the code. Connect it to the available network make sure it's connected to the same network as the computer on the output end.

After Connecting once there wouldn't be any need of establishing the connection you can use "sudo config" command on the raspberry terminal to take the IP address. The details of this step are mentioned in the VNC Viewer section. Use the link below if "sudo cofig" doesn't give the relevant information.

[6 Ways to Find a Raspberry Pi's IP Address \(makeuseof.com\)](https://makeuseof.com/6-ways-to-find-a-raspberry-pi-ip-address/)

RS232 CABLE:



According to the program created Pin 2, Pin 3 and Pin 5 will be used to establish the connection.

Caution: any form of cross connection can damage the Arduino, please ensure all the connections are proper before powering up the Arduino.

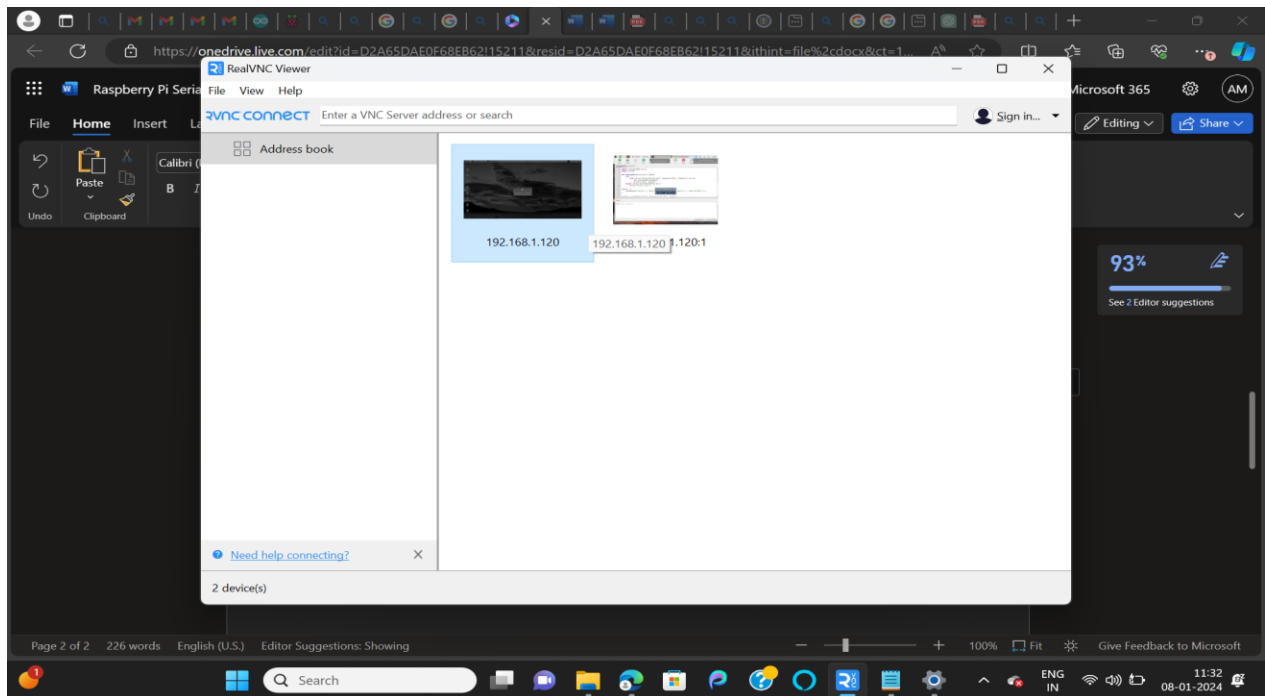
Arduino IDE:

Use Putty if the IDE doesn't recognize the device to make sure the signal is generating through Arduino. Go to **tools** select the "Arduino Uno" module from the **boards menu** and select **COM5** port and open Serial monitor (ctrl+shift+m). As soon as it recognizes COM5 choose the serial port in the dialogue box and adjust baud rate to the value mentioned in the prog5.py (115200 or 9600) in the dialogue box and continue. After writing code in the Arduino, upload it to the device by clicking the arrow on the left side window.

The Arduino is just a tool in this project to pass the characters using RS232 cable and display it. None of the processing is happening at the Arduino end. Since the connection is novel and Pi doesn't have a serial port, the characters are improperly recognized on the VS code.

Connection to the VNC Viewer:

Use Command **Sudo Config** to open the IP address of the Pi. It will be in the form **192.168.x.x**.



Enter the address after connecting to same network, it will ask for login credentials for **"aamishra"**

The **password is 122627.**

The virtual viewer will view the Raspberry pie screen after which the monitor will not be necessary.

VS code:

Open VS code and start decodeandemulate.py. After this the display will appear with hexadecimal characters. The output will be the characters that are being entered at the Pi Prog5.py. Each character will be recognized individually.

The characters being recognized here have been replaced or mapped (serial_mappings.csv), so that the data entered is processed correctly. However, digit "8" and letter "h" are not recognized at all and do not process any output on the screen. If decoder is used the values are unrecognizable in most cases.

Professor's suggested Enhancement 1:

Make an entire word recognize at once or send an entire word at once: I suggest putting some kind of sleep function which can wait for the characters a few seconds and then process the entire word.

The screenshot shows a Visual Studio Code editor window titled "Project_RaspberryPi". The Explorer sidebar on the left lists files: `charactermapping.py`, `cleanup.py`, `decodeandemulate.py` (selected), `decoded.py`, `EmulateKeyBoardPress.py`, `serial_mappings.csv`, and `Serial.py`. The main editor displays the code in `decodeandemulate.py`:

```
34 mapped_value = None
35
36 for row in csv_reader:
37     if row['Received_Value'] == str(data):
38         mapping_found = True
39         mapped_value = row['Mapping']
40         break
41
42 if mapping_found:
43     print(f"Mapped value: {mapped_value}")
44     pyautogui.press(str(mapped_value))
45 else:
46     print("Mapping not found for the received value.")
47
48
49 # last_received_data = data
50
51 except KeyboardInterrupt:
52     # Close the serial port when the program is terminated
53     ser.close()
```

Below the code editor is the TERMINAL panel, which shows the following output:

```
Mapped value: & C:/Users/aalek/AppData/Local/Programs/Python/Python311/python.exe c:/Project_RaspberryPi/decoded.py
Received data (hex): b''
Mapped value: & C:/Users/aalek/AppData/Local/Programs/Python/Python311/python.exe c:/Project_RaspberryPi/decoded.py
Received data (hex): b''
Mapped value: & C:/Users/aalek/AppData/Local/Programs/Python/Python311/python.exe c:/Project_RaspberryPi/decoded.py
Received data (hex): b''
Mapped value: & C:/Users/aalek/AppData/Local/Programs/Python/Python311/python.exe c:/Project_RaspberryPi/decoded.py
Received data (hex): b''
Mapped value: & C:/Users/aalek/AppData/Local/Programs/Python/Python311/python.exe c:/Project_RaspberryPi/decoded.py
```

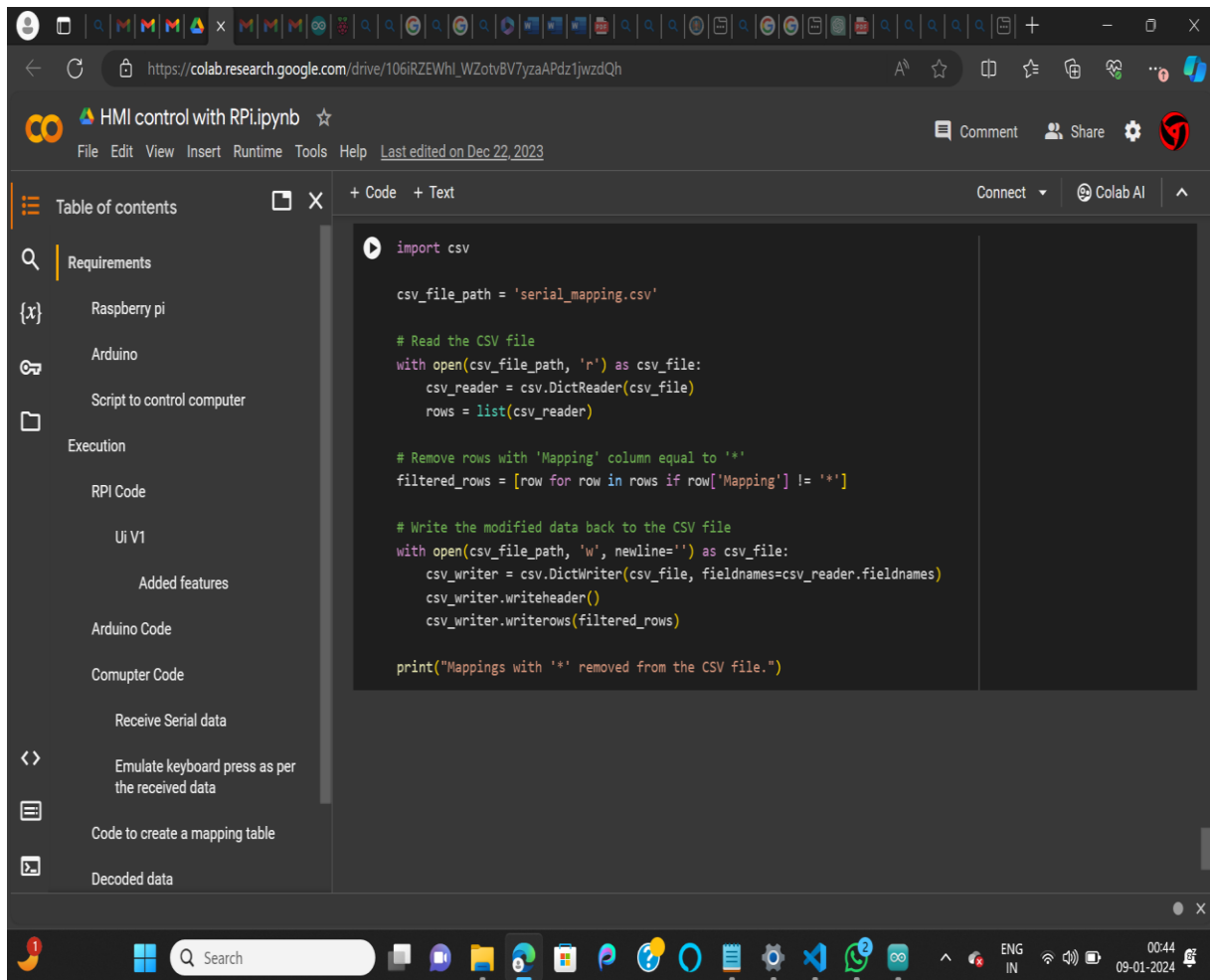
The terminal also shows a "History restored" message and the current directory path: `PS C:\Project_RaspberryPi>`. The status bar at the bottom indicates the file is at line 49, column 13, with 4 spaces, UTF-8 encoding, CRLF line endings, and Python 3.11.4 64-bit.

Professor's suggested Enhancement 2:

The signal transmission should be in both directions. From raspberry pi to Laptop and laptop to raspberry Pi. The HMI should be created in such a way that the progged or submitted command should transmit perfectly.

For New Mappings:

Use this code to create new mappings. Once created, do not run this code again.



```
import csv

csv_file_path = 'serial_mapping.csv'

# Read the CSV file
with open(csv_file_path, 'r') as csv_file:
    csv_reader = csv.DictReader(csv_file)
    rows = list(csv_reader)

# Remove rows with 'Mapping' column equal to ''
filtered_rows = [row for row in rows if row['Mapping'] != '']

# Write the modified data back to the CSV file
with open(csv_file_path, 'w', newline='') as csv_file:
    csv_writer = csv.DictWriter(csv_file, fieldnames=csv_reader.fieldnames)
    csv_writer.writeheader()
    csv_writer.writerows(filtered_rows)

print("Mappings with '' removed from the CSV file.")
```

Output:

Upon the use of any application like notes or notepad. The output will be processed in the form of keyboard emulation i.e. you can type the word which you processed from the Pi end.