Dorottya Kiss
Zsófia Schramek
Bálint Stellek

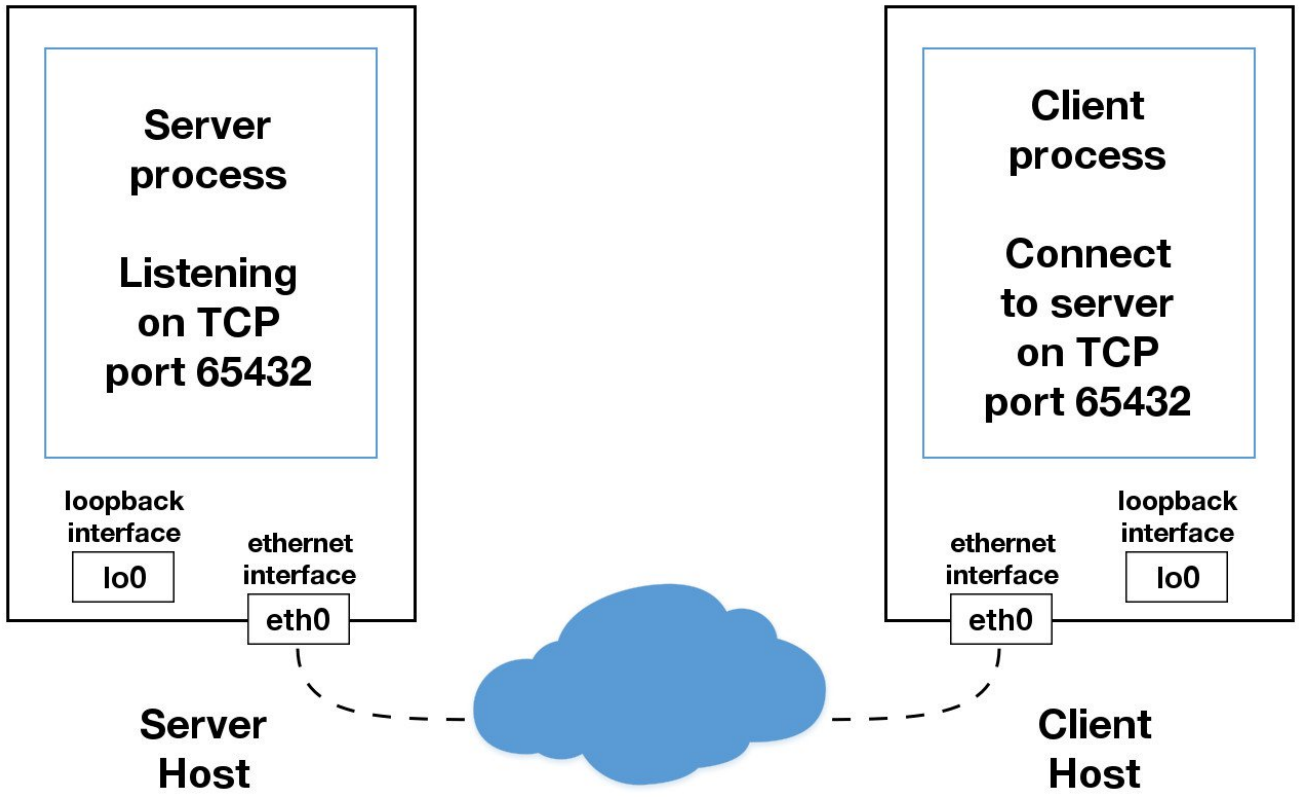
LAN communication

LAN communication description

Implement a client-server socket communication between the Raspberry Pi and a host PC. Write both sides of the program. Read the joystick of the Raspberry PI and send it over to the Host, where it is displayed graphically.

Socket communication

- Sockets and the socket API are used to send messages across a network.
- The most common type of socket applications are client-server applications, where one side acts as the server and waits for connections from clients.



```python
# Echo client program
import socket
import sys

import random
import time
from tkinter import *

def close_window ():
    root.destroy()

def socket_comm():
    HOST = '169.254.140.239'    # The remote host
    PORT = 50007        # The same port as used by the server
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.connect((HOST, PORT))
        while True:
            data = s.recv(1024)
            event = data.decode('utf-8')
            dirs.set(event)
            root.update_idletasks()
            root.update()
            time.sleep(0.2)
            # print(direction)

```

```python
# Echo server program
import socket
from sense_hat import SenseHat
from time import sleep

sense = SenseHat()

HOST = ''                 # Symbolic name meaning all available interfaces
PORT = 50007              # Arbitrary non-privileged port
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen(1)
    conn, addr = s.accept()
    with conn:
        print('Connected by', addr)
        while True:
            event = sense.stick.wait_for_event()
            conn.send(event.direction.encode())
            sleep(0.1)
```

Python Code

Graphical User Interface

- We had trouble with the Qt framework's QProcess() class
  - Couldn't call the socket script and receive multiple packages
- We created our GUI in TkInter which is Python's de-facto standard GUI (Graphical User Interface) package.

GUI code

```
26  root = Tk()
27  root.geometry("500x300")
28  root.resizable(0, 0)
29
30  root.minsize(500,300)
31  label = Label(root,text='Joystick GUI',bg = "gray")
32  label.config(width=100,  height=2, font=("Courier", 24, "bold"))
33  label.place(x=250, y=100, anchor="center")
34  label.pack()
35
36  root.config(bg='gray')
37  dirs = StringVar()
38  label2 = Label(root, textvariable=dirs, bg = "gray", borderwidth=2, relief="groove", width = 200)
39  label2.config(width=10, font=("Courier", 44))
40  label2.place(x=250, y=150, anchor="center")
41  dirs.set("Direction")
42
43
44  startbutton = Button(root, fg = "black", text = 'Start', bg = "gray", borderwidth=2, relief="raised", command = socket_comm)
45  startbutton.config(width=15, height=1, font=("Courier", 10, "bold"))
46  startbutton.place( x=200, y=270, anchor=SE)
47
48  stopbutton = Button(root, fg = "black", text = 'Quit', bg = "gray", borderwidth=2, relief="raised", command = close_window)
49  stopbutton.config(width=15, height=1, font=("Courier", 10, "bold"))
50  stopbutton.place( x=300, y=270, anchor=SW)
51
52  root.mainloop()
```