**Solution to Communication Subsystem**

Communication subsystem enables the robot to communicate with the opponent using the handshaking. According to the standard committee, Wi-Fi modules must be used to implement handshaking. Since Raspberry Pi was used in the project, there is no need to get a separate Wi-fi module; the internal Wi-fi module of the Raspberry Pi was used.

Socket programming is an effective tool to implement client-server communication algorithms. It can be implemented in Python or C++. Our algorithms are written in Python for now, yet it can easily be converted to C++ if the team members decide it is necessary. The algorithms for client and server sides are slightly different. Figure … shows the functions that are used for client and server sides to create communication between client and server.

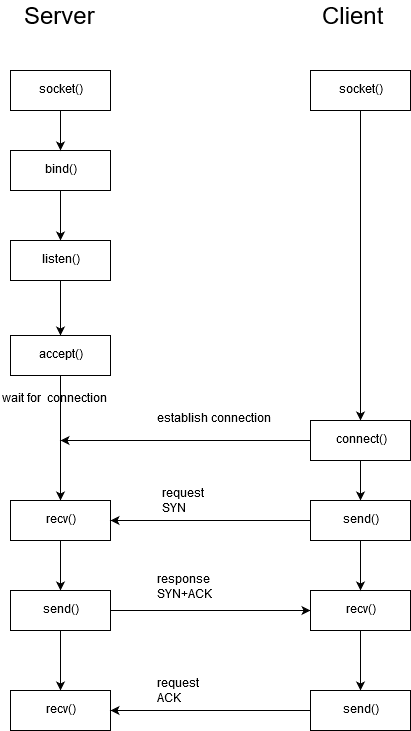


Figure : Basic Functions in Python Socket Programming to Implement Handshaking

Here is the summary of the key functions from socket library:

* **socket.socket()**: Creates a new socket using the given address family, socket type and protocol number.
* **s.bind(address)**: Binds the socket to the address defined previously.
* **s.listen(backlog):** Sets up the maximum number of connections that can be made to the socket, which must be at 1 for the project.
* **s.accept()**: Waits until connection arrives, than accept the client connection. Returns the client socket connected to the server as **(conn, address)** pair, where conn is a new socket object and address is the address bound to this socket
* **s.connect():** Provides client to connect to the server
* **s.send():** Transmits message to the remote socket.
* **s.recv():** Receives message from the remote socket
* **socket.close()**: closes the socket; i.e., ends the communication with the opponent at the end of the race.

It is stated in the standard committee that each team must be assigned a static IP to communicate with the other robots. Duayenler has the static IP stated as “192.168.1.7” and the ID as “07”. Since Raspberry Pi 3 comes with a built-in wireless adapter, configuring it as a Wi-Fi hotspot is possible. To assign given IP to the robot, Raspberry Pi must be set as an access point from the terminal.

In the algorithm that was implemented for the handshake, in a continuous loop, the front and rear sensors’ values are been checked. There are two functions which are for client and server modes, respectively. If the front sensor senses the opponent in 5 cm range, our main code visits the client mode function. If the rear sensor senses the opponent in 5 cm range, server mode function runs. If our robot is in the server mode, the rear sensor value is again checked. The acknowledge message(<ID>01) or reject message(<ID>11) is sent according to the sensor value.

**Test Results to Communication Subsystem**

The codes that were written in Python were tested in 3 different combinations. The first and simplest test has been done on one computer (or raspberry pi) using the same device as client and server, at the same time. To achieve that, the computer’s (or raspberry pi’s) IP address should be defined in the host section defined in the client mode function. Secondly, the codes were tested on two computers. Thirdly, one raspberry pi and one computer were used for the test. All tests were successful if the server side is connected to the internet and client side is connected to the server via hotspot. The outputs of the tests were given in the figure …

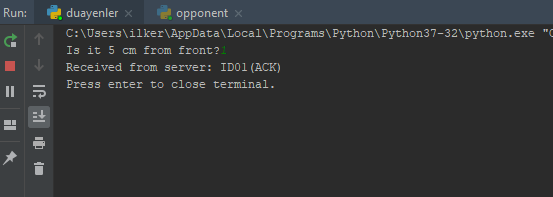


Figure : Test Results of Handshaking for Client Side

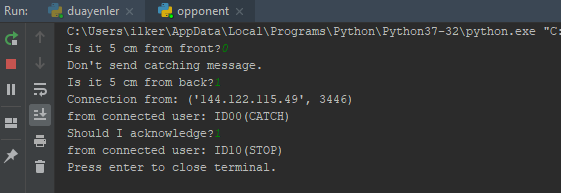


Figure : Test Results of Handshaking for Server Side