Laboratory – Ethernet

How the program works

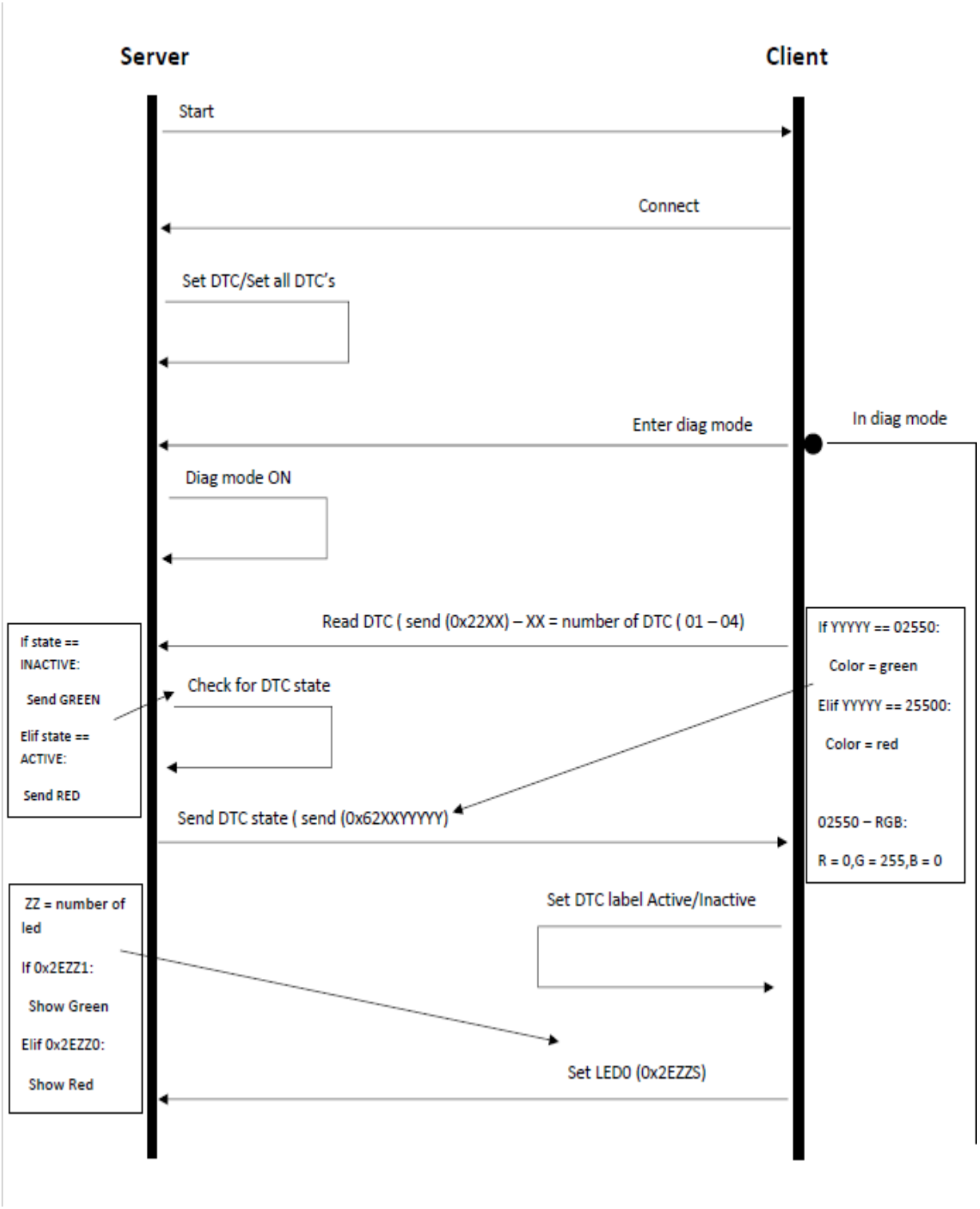
The server starts and the client connects to it. ‘SET DTC’ button sets 4 leds to **active/inactive**. (ACTIVE = Red, INACTIVE = Green). ‘Set all DTC’ button set all DTC’s active/inactive.

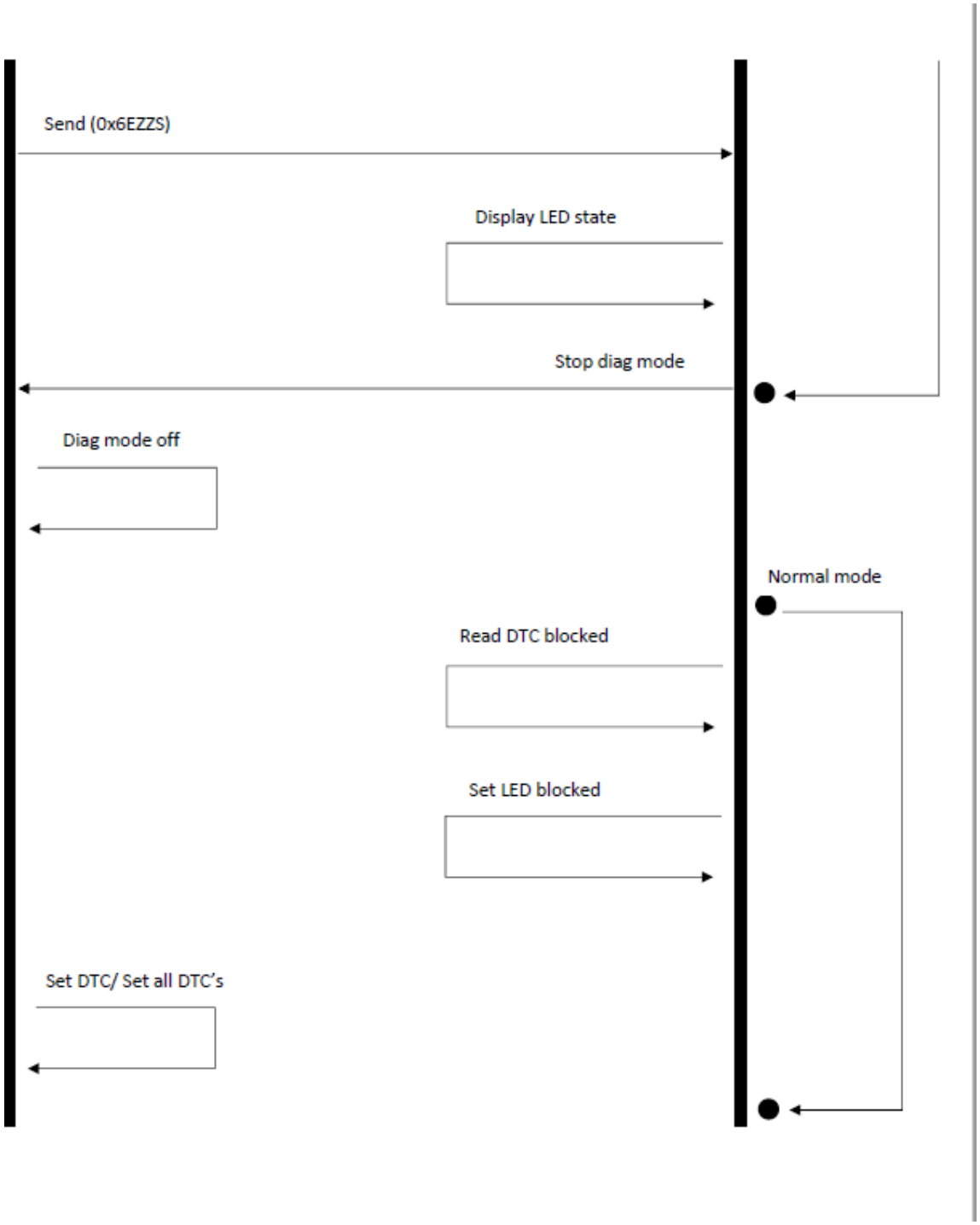
‘SET LED’ button on client side sends a request to the server to set a led color to Green/Red on client GUI.

‘Enter diag mode’ button start the diagnosis mode and **‘Stop diag mode’** stops the diagnosis mode. In **diagnosis mode** the client can communicate with the server by setting leds on it or read DTC’s status.

In DIAG MODE:
The server can control the DTC’s and the client can READ DTC’s status. The client
can set LED states and the server executes client requests.

In Normal mode (DIAG MODE off):
The server still can control the DTC’s but does not communicate with the client


Server
Client
Start
Connect
Set DTC/Set all DTC's
Enter diag mode
In diag mode
Diag mode ON
Read DTC (send (0x22XX) – XX = number of DTC (01 – 04)
If state == INACTIVE:
Send GREEN
Elif state == ACTIVE:
Send RED
Check for DTC state
If YYYYY == 02550:
Color = green
Elif YYYYY == 25500:
Color = red
02550 – RGB:
R = 0,G = 255,B = 0
Send DTC state (send (0x62XXYYYYY)
ZZ = number of led
If 0x2EZZ1:
Show Green
Elif 0x2EZZ0:
Show Red
Set DTC label Active/Inactive
Set LED0 (0x2EZZS)


Send (0x6EZZS)
Display LED state
Stop diag mode
Diag mode off
Normal mode
Read DTC blocked
Set LED blocked
Set DTC/ Set all DTC's

- **Exercise 0**

Complete in **server-gui.py start_server** function in order to create a server using method described in previous laboratories.

Complete in **client.py start_client** function with necessary code in order to connect to the server using the method described in previous laboratories.

- **Exercise 1**

Complete in **client.py**, **diag** function with necessary code in order to send to the server ‘0x3E01’ and set a variable on 1, this means that the diag mode is on.

Complete in **client.py**, **stop_diag** function in order to send ‘0x3E00’ and set a variable on 0, this means that the diag mode is off.

Complete in **server-gui.py**, **recv_handler** function in order to check if the diag mode is ON/OFF.

- **Exercise 2**

Complete in **server-gui.py**, **set_dtcX** function in order to set the DTC state active/inactive when **Set DTC X** button is pressed.

Complete in **server-gui.py**, **set_all** function in order to set all the DTC state active/inactive when **Set all DTC** button is pressed.

- **Exercise 3**

Complete in **client.py**, **get_dtc_state** function in order to send, only in **diag mode**, a hex number ‘0x22XX’ (XX number of dtc – 01 – 04).

Complete in **server-gui.py**, **read_dtcX** functions in order to send a message ‘0x62XXYYYYY’ (See the diagram for YYYYY explanation).

- **Exercise 4**

Complete in **client.py**, **set_ledX_flags** function in order to send, only in diag mode, ‘0x2EZZS’ (See the explanation in the diagram).

Complete **server-gui.py**, **set_ledX**, functions in order to send, when in diag mode, the response for ‘0x2EZZS’ and display the state on the leds.

Response for ‘0x2EZZS’ is ‘0x6EZZS’.

Complete in **client.py**, **set_ledX_label** function in order to decode the response message and to show the LED state on client Gui.