Client

1.Client reads 2-4 command line arguments

Mandatory

-IP address of server

-Port of server

Optional

-Integer used to communicate if the lightbulb should turn on or off(1 for on, 2 for off)

-String used to communicate the color change of the lightbulb

-If the optional integer and string arguments are provided the lightbulb should switch on and change color

-If only the optional integer argument is provided the lightbulb should only turn off or remain off

-If only the optional string argument is provided the lightbulb should only change its color.

2.Send a request with the command line arguments of the ip address and port of the server using the message format shown below

3.Wait for a response using a 1 second timeout period.

-If a response arrives with the timeout period, print out the server response as shown below

-If no response retry for a maximum of three times before printing an error message and exiting

Server

1.Server reads 2 command line arguments

-IP address of server

-Port of server

2.Server responds to request from client

-Turns on the lightbulb(if applicable)

-Changes the light color(if applicable)

-Turns off the lightbulb(if applicable)

-Returns the status of the lightbulb

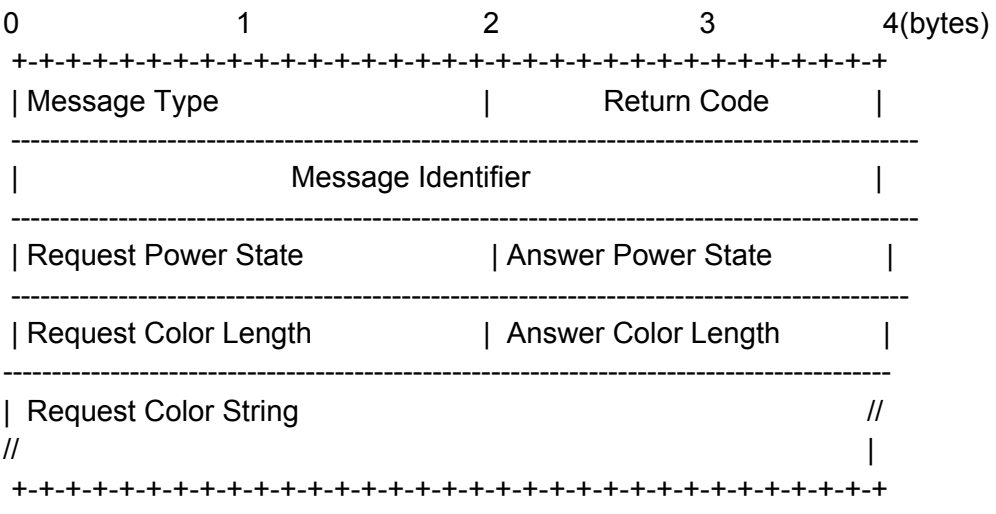
3.Returns an error code based on these conditions

-Lightbulb won't turn on(1)

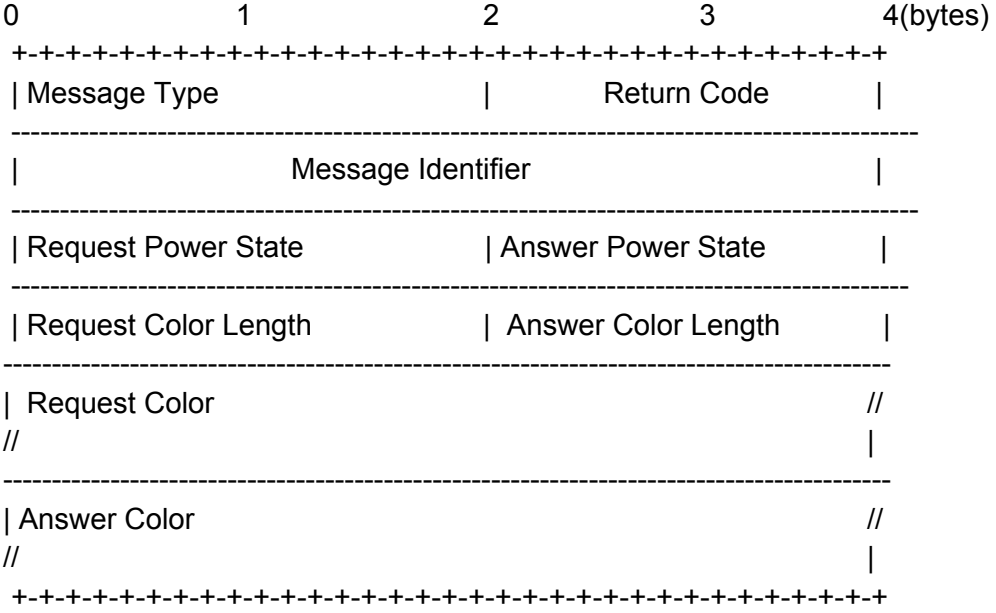
-Color not supported(2)

-Message format not supported(for example the power state of the lightbulb uses 0, 1, and 2 but a 3 was given)

Client to Server



Server to Client



Message Type(16 bits): 1 on request, 2 on response
Return Code(16 bits): 0 on request. For response 0 if no errors, 1 if bulb wont turn on, 2 if color is not supported, 3 if the message format is incorrect
Message Identifier(32 bits): Unique identifier that is sent to server and server echos same number back. Random generated number in range between 1-100
Request Power State(16 bits):0 for default(no change), else integer given in command line argument
Answer Power State(16 bits):0 in request because power state is unknown, for response 1 if lightbulb is on, 2 if lightbulb is off
Request Color Length(16 bits):In request length is used to carry string for the type of color,length is 0 if color change is not specified.

Answer Color Length(16 bits):In request length is 0 because no answer, on response length is used to carry the string of the bulb's active color.
Request Color String(variable length):In request , string is the color that the lightbulb should be changed to if specified.
Answer Color String(variable length):In request there is no answer color string, in response the server sends the color of the active lightbulb color setting.

Test Output
Test Case 1: Client Output (Turning on the lightbulb and changing the color)

Sending Request to 127.0.0.1, 9999:
Message ID: 23
Request Power State: 1
Request Color Length: 4 bytes
Request Color: blue

Received Response from 127.0.0.1, 9999:
Return Code 0 (No Errors)
Message ID: 23
Request Power State: 1
Request Color Length: 4 bytes
Request Color: blue
Answer Power State : 1
Answer Color Length: 4 bytes
Answer Color: blue

Test Case 2: Client Output (Changing color of lightbulb)

Sending Request to 127.0.0.1, 9999:
Message ID: 24
Request Power State: 0
Request Color Length: 4 bytes
Request Color: blue

Received Response from 127.0.0.1, 9999:
Return Code 0 (No Errors)
Message ID: 24
Request Power State: 0
Request Color Length: 4 bytes
Request Color: blue
Answer Power State : 1
Answer Color Length: 4 bytes
Answer Color: blue

Test Case 3: Client Output(Determine the status of the light)

Sending Request to 127.0.0.1, 9999:
Message ID: 25
Request Power State: 0
Request Color Length: 0 bytes
Request Color:

Received Response from 127.0.0.1, 9999:
Return Code 0 (No Errors)
Message ID: 25
Request Power State: 0
Request Color Length: 0 bytes
Request Color:
Answer Power State : 1
Answer Color Length: 4 bytes
Answer Color: blue

Test Case 4: Client Output(Turn the light off)

Sending Request to 127.0.0.1, 9999:
Message ID: 26
Request Power State: 2
Request Color Length: 0 bytes
Request Color:

Received Response from 127.0.0.1, 9999:
Return Code 0 (No Errors)
Message ID: 26
Request Power State: 2
Request Color Length: 0 bytes
Request Color:
Answer Power State : 2
Answer Color Length: 4 bytes
Answer Color: blue

Test Case 5: Client Output(bulb is not functioning)

Sending Request to 127.0.0.1, 9999:
Message ID: 26
Request Power State: 1
Request Color Length: 4 bytes
Request Color: blue

Received Response from 127.0.0.1, 9999:
Return Code 1 (bulb is not functioning)
Message ID: 26
Request Power State: 1
Request Color Length: 4 bytes
Request Color: blue

Test Case 6: Client Output(Color not supported)

Sending Request to 127.0.0.1, 9999:
Message ID: 27
Request Power State: 0
Request Color Length: 4 bytes
Request Color: buel

Received Response from 127.0.0.1, 9999:
Return Code 2 (color not supported)
Message ID: 27
Request Power State: 0
Request Color Length: 4 bytes
Request Color: buel

Test Case 7: Client Output(message format not supported)

Sending Request to 127.0.0.1, 9999:
Message ID: 28
Request Power State: 636475
Request Color Length: 4 bytes
Request Color: blue

Received Response from 127.0.0.1, 9999:
Return Code 3 (message format not supported)
Message ID: 28
Request Power State: 636475
Request Color Length: 4 bytes
Request Color: blue

Test Case 8: Client Output(no response)

Sending Request to 127.0.0.1, 9999:
Message ID: 88
Request Power State: 0
Request Color Length: 4 bytes
Request Color: blue
Request timed out …
Sending Request to 127.0.0.1, 9999:
Request timed out …
Sending Request to 127.0.0.1, 9999:
Request timed out … Exiting Program.