

```

1  import socket
2  import struct
3
4  # Global Variables
5  host = None
6  port = None
7  data = None
8
9
10 # UDP Checksum Function
11 def checksum_func(data):
12     checksum = 0
13     data_len = len(data)
14
15     # Appends 0's to the end of data and adjusts data_len
16     if (data_len % 2):
17         data_len += 1
18         data += struct.pack('!B', 0)
19
20     # Compute the sum
21     for i in range(0, data_len, 2):
22         w = (data[i] << 8) + (data[i + 1])
23         checksum += w
24
25     # Wrap around bit
26     checksum = (checksum >> 16) + (checksum & 0xFFFF)
27
28     # Complement the result
29     checksum = ~checksum & 0xFFFF
30     return checksum
31
32
33 # Create Socket
34 def socket_create():
35     global host
36     global port
37     global s
38     host = "localhost"
39     port = 1001
40
41     try:
42         s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
43         s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
44     except socket.error as msg:
45         print("Socket creation error: " + str(msg))
46
47
48 # State0
49 def state0():
50     global host
51     global port
52     global s
53     global data
54
55     message = input("Message: ")
56     if message == "quit":
57         return None
58     else:
59         try:
60             # Compute checksum
61             checksum = checksum_func(bytes(message.encode('utf-8')))
62
63             # Append checksum to data
64             data = str(message) + "|" + str(checksum)
65             s.sendto(data.encode('utf-8'), (host, port))
66

```

```

67         except socket.error as msg:
68             print("Error sending message: " + str(msg))
69
70     return state1
71
72
73 # State 1
74 def statel():
75     global host
76     global port
77     global s
78     global data
79
80     try:
81
82         print("Waiting for ACK or NACK...")
83
84         response, addr = s.recvfrom(1024)
85
86         answer = str(response.decode('utf-8'))
87         if answer == "NACK":
88             # Resend message
89             s.sendto(data.encode('utf-8'), (host, port))
90             print("Sending message again...")
91             return state1
92
93         elif answer == "ACK":
94             print("Message received.")
95             return state0
96
97     except socket.error as msg:
98         print("Error sending message: " + str(msg))
99
100
101 # Main Function
102 if __name__ == "__main__":
103     global s
104
105     # created socket
106     socket_create()
107
108     # Initial State
109     state = state0
110     while state:
111         state = state()
112
113     s.close()
114     print("FSM Done")

```