Setting Command of ultrasonic sensor

Baud rate: 9600

Note: If you want to setting a ultrasonic sensor, can only connect one ultrasonic sensor to the 485 interface. You cannot set several ultrasonic sensors together.

1. The ultrasonic sensor has four working mode: sensor, led display, factory mode, wireless sensor. Regardless of the type of work, it can be modified by the following command via the 485 interface.

Set as ultrasonic sensor:

00 FF 01 00 01 00 14 63

Set as screen:

00 FF 01 00 01 01 D5 A3

1. The address code of ultrasonic sensor can be modified by the 485 interface with the following instructions.

00 FF 01 04 01 01 94 62

00 FF 01 04 01 02 D4 63

00 FF 01 04 01 03 15 A3

00 FF 01 04 01 04 54 61

00 FF 01 04 01 05 95 A1

00 FF 01 04 01 06 D5 A0

00 FF 01 04 01 07 14 60

00 FF 01 04 01 08 54 64

00 FF 01 04 01 09 95 A4

00 FF 01 04 01 0A D5 A5

00 FF 01 04 01 0B 14 65

00 FF 01 04 01 0C 55 A7

00 FF 01 04 01 0D 94 67

00 FF 01 04 01 0E D4 66

00 FF 01 04 01 0F 15 A6

00 FF 01 04 01 10 54 6E

00 FF 01 04 01 11 95 AE

00 FF 01 04 01 12 D5 AF

00 FF 01 04 01 13 14 6F

00 FF 01 04 01 14 55 AD

00 FF 01 04 01 15 94 6D

00 FF 01 04 01 16 D4 6C

00 FF 01 04 01 17 15 AC

00 FF 01 04 01 18 55 A8

00 FF 01 04 01 19 94 68

00 FF 01 04 01 1A D4 69

00 FF 01 04 01 1B 15 A9

00 FF 01 04 01 1C 54 6B

00 FF 01 04 01 1D 95 AB

00 FF 01 04 01 1E D5 AA

00 FF 01 04 01 1F 14 6A

The red number is the address, and the last two bits are the CRC check code, using the same check code algorithm as the MODBUS protocol.

3. As sensor, you can set offline or online mode. The instructions as follows:

Set offline mode: (sensor don not connect with controller)

00 FF 01 06 01 00 F4 62

Set online mode: (sensor connect with controller)

00 FF 01 06 01 01 35 A2

4. As sensor, you can read its parking status with the following command:

00 00 00 01 01 E4 50 //address code 0

00 01 00 01 01 E5 AC //address code 1

00 02 00 01 01 E5 E8 //address code 2

00 03 00 01 01 E4 14 //address code 3

00 04 00 01 01 E5 60 //address code 4

00 05 00 01 01 E4 9C //address code 5

00 06 00 01 01 E4 D8 //address code 6

00 07 00 01 01 E5 24 //address code 7

00 08 00 01 01 E6 30 //address code 8

00 09 00 01 01 E7 CC //address code 9

00 0a 00 01 01 E7 88 //address code 10

00 0b 00 01 01 E6 74 //address code 11

00 0c 00 01 01 E7 00 //address code 12

00 0d 00 01 01 E6 FC //address code 13

00 0e 00 01 01 E6 B8 //address code 14

00 0f 00 01 01 E7 44 //address code 15

00 10 00 01 01 E0 90 //address code 16

00 11 00 01 01 E1 6C //address code 17

00 12 00 01 01 E1 28 //address code 18

00 13 00 01 01 E0 D4 //address code 19

00 14 00 01 01 E1 A0 //address code 20

00 15 00 01 01 E0 5C //address code 21

00 16 00 01 01 E0 18 //address code 22

00 17 00 01 01 E1 E4 //address code 23

00 18 00 01 01 E2 F0 //address code 24

00 19 00 01 01 E3 0C //address code 25

00 1a 00 01 01 E3 48 //address code 26

00 1b 00 01 01 E2 B4 //address code 27

00 1c 00 01 01 E3 C0 //address code 28

00 1d 00 01 01 E2 3C //address code 29

00 1e 00 01 01 E2 78 //address code 30

00 1f 00 01 01 E3 84 //address code 31

The reply is: 00 XX 80 01 01 YY MM NN,

XX is the address

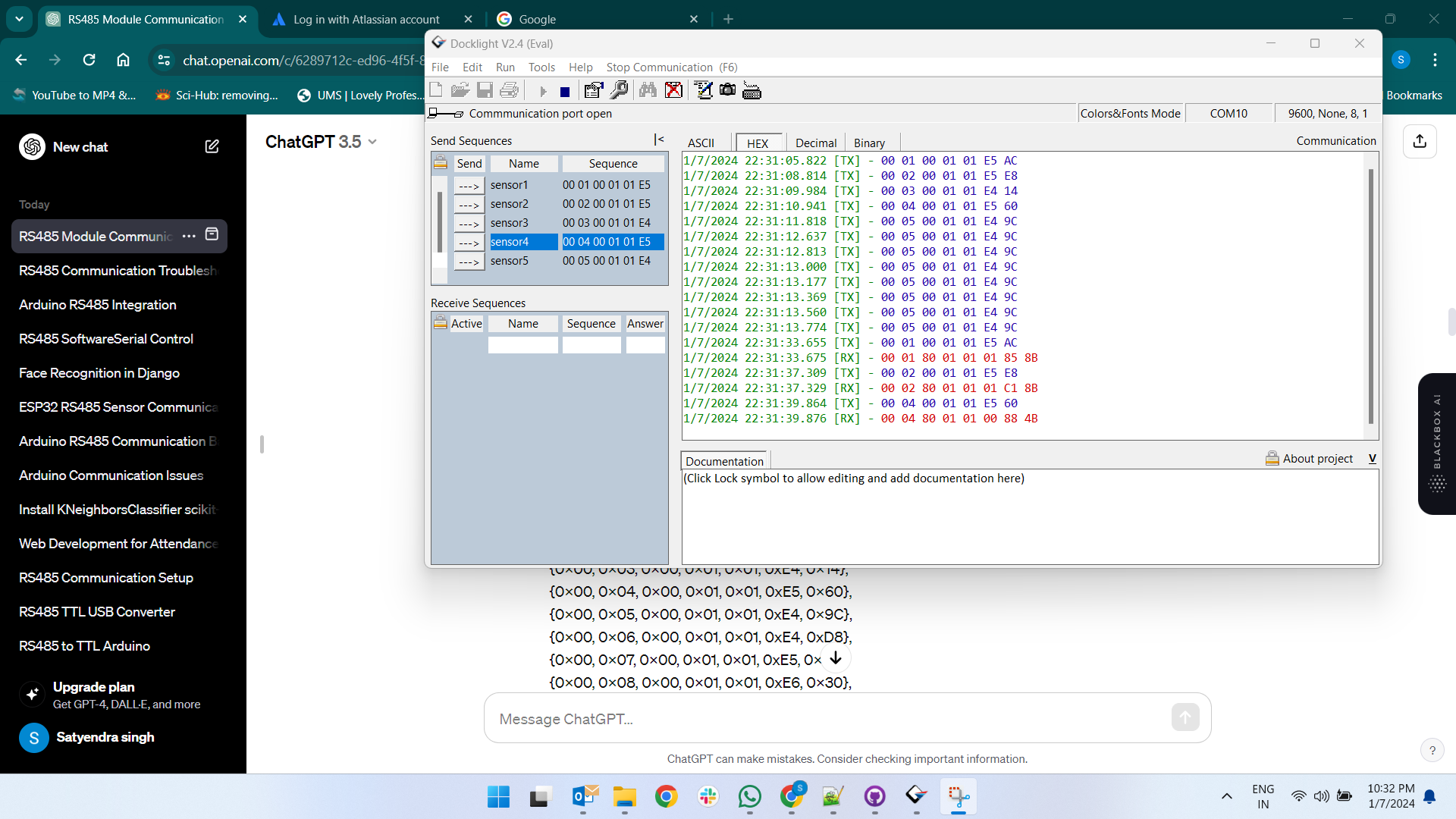
YY is the status of parking space.

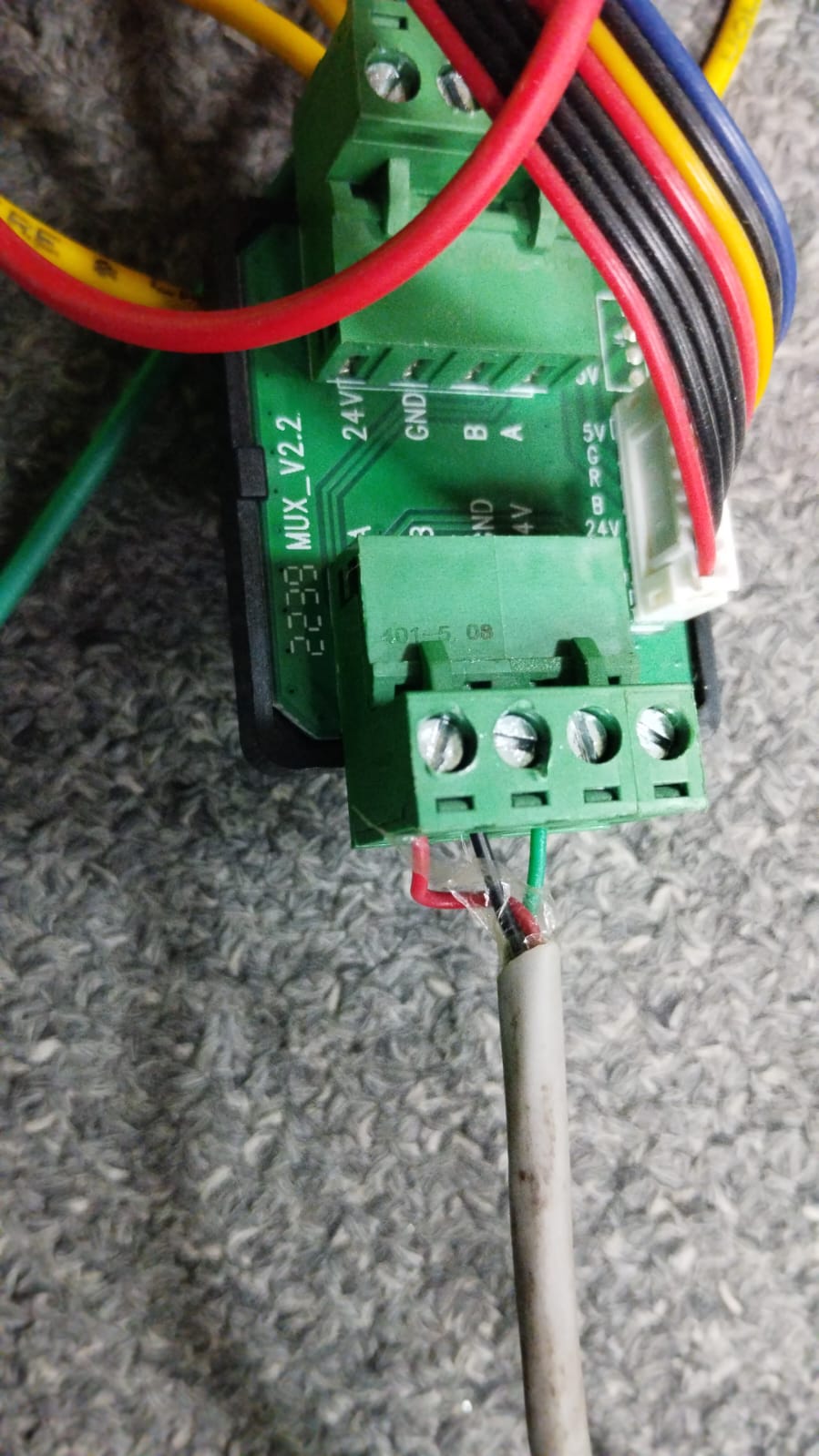
If there is a car, YY is 01, no car, YY is 00. MM NN is the check code.

Note- if we send the 7 byte data then we received the 8 byte data.. xx is the address, yy is the status, mm and NN are the crc check which should be the same for both transmission and receiver

00 1f 00 01 01 E3 84 this should be the reply we get from the sensor

00 XX 80 01 01 YY MM NN

Step- hum docklight serial software ka use krenge. Isme hum sabse pahle apna com port select krenge jo ki com10 hai. Baudrate 9600 rakhenge.uske bad name me click krenge sensor-1 likhenge aur hex me click karenge aur jo data send karna hai wo likhenge jaise 00 01 00 01 01 E5 AC isi tarah jitney bhi sensor connected hai us sensor ka naam with sensor data ke hisab se likh denge aur eke k krke data ko send karenge to hume udhar se sensorkaresponsemilega



Hum rs485 to usb ke cable ka use kiye hai. Jisme ki red wire A+ se black wire B- se aur gnd ko gnd se connect kiye hai. So humara communication sahi se kaam kar rha hai.

Note- so humne python ka code likha hai sensor ke response ko read karne ke liye

**import serial**

**import time**

**# Define serial port settings**

**port = 'COM10' # Update this with the correct port name on your system**

**baudrate = 9600 # Set the baud rate according to your device specifications**

**timeout = 1**

**# Create a serial connection**

**ser = serial.Serial(port, baudrate, timeout=timeout)**

**try:**

**if ser.isOpen():**

**print(f"Serial port {port} is open.")**

**# Create requests for 10 sensors (adjust as needed)**

**sensor\_requests = [**

**bytes.fromhex('00 00 00 01 01 E4 50'), # Sensor 0 request**

**bytes.fromhex('00 01 00 01 01 E5 AC'), # Sensor 1 request**

**bytes.fromhex('00 02 00 01 01 E5 E8'), # Sensor 2 request**

**bytes.fromhex('00 03 00 01 01 E4 14'), #Sensor 3 request**

**bytes.fromhex('00 04 00 01 01 E5 60'), # Sensor 4 request**

**bytes.fromhex('00 05 00 01 01 E4 9C')**

**]**

**# Send requests for each sensor**

**for index, request in enumerate(sensor\_requests, start=1):**

**ser.write(request)**

**print(f"Sent sensor {index} request:", request)**

**# Wait for a response (adjust this based on your device response time)**

**time.sleep(0.1)**

**# Read response**

**response = ser.read\_all()**

**if response:**

**print(f"Received response for sensor {index}:", response.hex())**

**else:**

**print(f"No response received for sensor {index}.")**

**else:**

**print(f"Could not open serial port {port}.")**

**except serial.SerialException as e:**

**print("Serial connection error:", e)**

**finally:**

**ser.close() # Close the serial port when finished**

Note- par hum chahte hai ki hum ye sensor ke data jo mil rha hai response me usse hum 6th byte ko extract kar le. Ye 6th byte hume sensor ka status btata hai ki sensor engaged hai ya disengaged. Agar sensor ka status 00 hai means sensor engaged hai agar data 01 aaya 6th byte ka means sensor disengaged hai. Aur man lo humare pass 5 sensors hai to calculate krenge availability space.

import serial

import time

# Define serial port settings

port = 'COM10' # Update this with the correct port name on your system

baudrate = 9600 # Set the baud rate according to your device specifications

timeout = 1

# Define the total number of parking spaces

total\_spaces = 5 # Change this value based on the actual number of parking spaces

# Create a serial connection

ser = serial.Serial(port, baudrate, timeout=timeout)

try:

if ser.isOpen():

print(f"Serial port {port} is open.")

# Create requests for 5 sensors (adjust as needed)

sensor\_requests = [

bytes.fromhex('00 00 00 01 01 E4 50'), # Sensor 0 request

bytes.fromhex('00 01 00 01 01 E5 AC'), # Sensor 1 request

bytes.fromhex('00 02 00 01 01 E5 E8'), # Sensor 2 request

bytes.fromhex('00 03 00 01 01 E4 14'), # Sensor 3 request

bytes.fromhex('00 04 00 01 01 E5 60') # Sensor 4 request

]

# Initialize variables to count available spaces

available\_spaces = total\_spaces

# Send requests for each sensor

for index, request in enumerate(sensor\_requests, start=0):

ser.write(request)

print(f"Sent sensor {index} request:", request)

# Wait for a response (adjust this based on your device response time)

time.sleep(0.1)

# Read response

response = ser.read\_all()

if response:

# Extract the 6th byte from the response

sixth\_byte = response[5] # Python uses 0-based indexing

# Check if the 6th byte is 0x00 (sensor engaged) or not

sensor\_status = "Engaged" if sixth\_byte == 0x00 else "Disengaged"

if sensor\_status == "Engaged":

available\_spaces -= 1 # Decrement available spaces if sensor is engaged

print(f"Received response for sensor {index}: Sensor is {sensor\_status}")

else:

print(f"No response received for sensor {index}.")

print(f"Available parking spaces: {available\_spaces}/{total\_spaces}")

else:

print(f"Could not open serial port {port}.")

except serial.SerialException as e:

print("Serial connection error:", e)

finally:

ser.close() # Close the serial port when finished

Example-1 for aduino code to capture the response using rs485 to ttl