**Nymbel assessment**

**Firmware task:**

**PC to MCU:**

**Code:**

1. The below code will work as
2. **Setup Function:**
   * Initializes the serial communication at a baud rate of 2400 bits per second.
3. **Loop Function:**
   * Continuously checks for incoming data from the PC.
   * Stores the received characters in an array (**receivedText**) until it receives the entire text (1000 characters including spaces).
   * Once the entire text is received, it sends it back to the PC.

The Arduino continuously listens for incoming characters from the PC. Once it receives the entire text, it starts transmitting the received text back to the PC.

**#include <EEPROM.h>**

**#define BAUD\_RATE 2400**

**#define TEXT\_SIZE 1000**

**void setup() {**

**Serial.begin(BAUD\_RATE);**

**}**

**void loop() {**

**static char receivedText[TEXT\_SIZE];**

**static int textIndex = 0;**

**static bool textReceived = false;**

**// Receive text from PC**

**while (Serial.available() && !textReceived) {**

**char receivedChar = Serial.read();**

**receivedText[textIndex++] = receivedChar;**

**if (textIndex >= TEXT\_SIZE - 1) {**

**receivedText[textIndex] = '\0'; // Null-terminate the string**

**textReceived = true;**

**}**

**}**

**// Transmit received text back to PC**

**if (textReceived) {**

**int addr = 0;**

**while (true) {**

**char sendChar = receivedText[addr];**

**if (sendChar == '\0') {**

**break; // End of string**

**}**

**Serial.write(sendChar);**

**addr++;**

**}**

**textReceived = false;**

**textIndex = 0;**

**}**

**}**

**MCU to PC:**

**MCU to PC:**

**Code:**

**The below code works as**

1. **Serial Port Initialization:**
   * Opens the serial port connection with the specified port (replace **'COMX'** with your Arduino's port).
2. **Serial Port Configuration:**
   * Configures the serial port settings like baud rate (2400), data bits, etc.
3. **Data Transmission from PC to MCU:**
   * Sends the text to the Arduino via the serial port.
   * Measures the time taken for the transmission and calculates the speed in bits per second.
4. **Data Reception from MCU to PC:**
   * Receives the text transmitted by the Arduino via the serial port.
   * Measures the time taken for the reception and calculates the speed in bits per second.
   * Prints the received text on the console.

#include <iostream>

#include <Windows.h>

#include <chrono>

int main() {

HANDLE serialPort = CreateFile(

"COMX", // Replace 'COMX' with your Arduino's port

GENERIC\_READ | GENERIC\_WRITE,

0,

NULL,

OPEN\_EXISTING,

FILE\_ATTRIBUTE\_NORMAL,

NULL

);

if (serialPort == INVALID\_HANDLE\_VALUE) {

std::cerr << "Error opening serial port\n";

return 1;

}

DCB dcbSerialParams = { 0 };

dcbSerialParams.DCBlength = sizeof(dcbSerialParams);

if (!GetCommState(serialPort, &dcbSerialParams)) {

std::cerr << "Error getting serial port state\n";

CloseHandle(serialPort);

return 1;

}

dcbSerialParams.BaudRate = CBR\_2400;

dcbSerialParams.ByteSize =