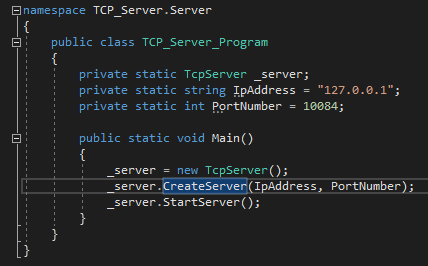
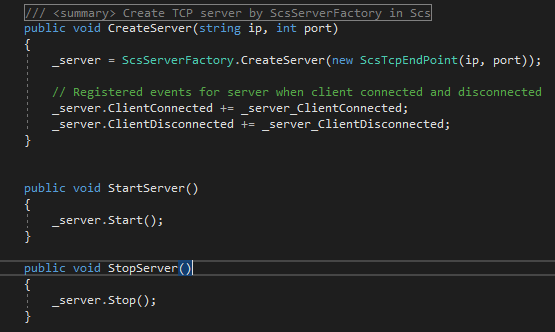
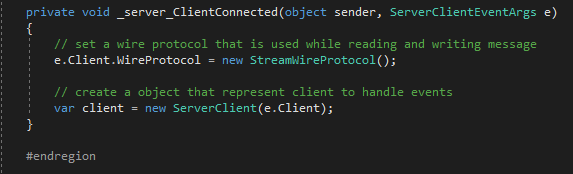
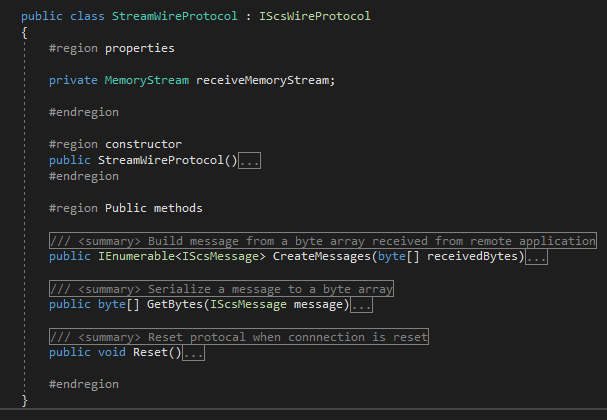
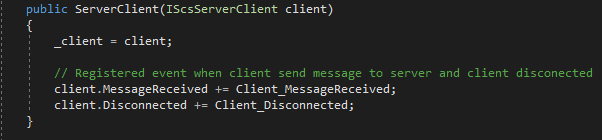
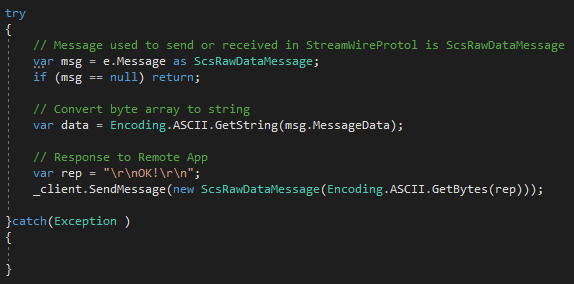
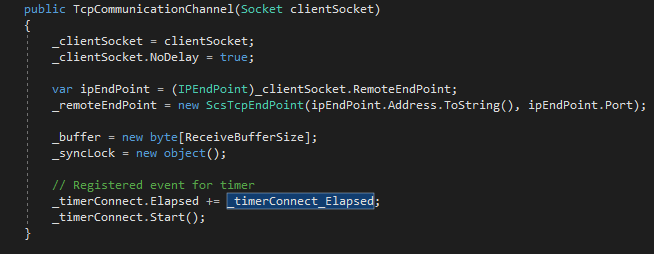
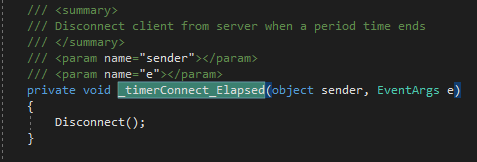
**BÁO CÁO TUẦN**

1. ***Công việc hoàn thành:***
   * Tạo ra một Web API project, khi chạy tạo ra một TCP Server.
   * Server mở kết nối cho client kết nối lên.
   * Server nhận dữ liệu của client gửi lên và phản hồi lại cho client.
   * Giải phóng connect sau một thời gian client không gửi dữ liệu lên.
2. ***Cấu trúc chương trình:***Solution gồm 2 project:
   1. **Scs**: thư viện Scs.
   2. **TCP\_Server**: Web API với 4 class.
      * **TCP\_Server\_Program**: chương trình tạo ra đối tượng Server và start Server để Client kết nối lên.
      * **TcpServer**: lớp đối tượng Server.
      * **ServerClient**: lớp đối tượng Client biểu diễn cho một Client khi kết nối đến Server.
      * **StreamWireProtocol**: lớp cài đặt giao thức dùng để giao tiếp qua lại giữa Client và Server.
3. ***Mô tả hoạt động:***
   * Ban đầu địa chỉ Ip và Port được gán cứng, ta tạo ra đối tượng **TcpServer**, tạo Server từ Ip và Port và start Server đó để Client kết nối đến.  
     
   * Trong lớp **TcpServer**, ta tạo Server bằng lớp **ScsServerFactory** trong thư viện **Scs**, cung cấp tham số đầu vào là địa chỉ Ip và Port. Sau đó đăng ký hai sự kiện cho Server khi Client kết nối tới và Client ngắt kết nối.  
     
   * Trong hàm xử lý sự kiện Client kết nối tới của Server, ta gán giá trị **WireProtocol** của Client kết nối tới bằng một đối tượng **StreamWireProtocol** quy định các thao tác khi trao đổi dữ liệu giữa Client và Server. Sau đó tạo ra một đối tượng **ServerClient** là đại diện của Client khi kết nối tới Server.  
     
   * Đối tượng **StreamWireProtocol** được ta implement từ Interface **IScsWireProtocol** trong thư viện **Scs**.   
     
   * Khi tạo ra đối tượng của lớp **ServerClient**, ta đăng ký sự kiện khi Client gửi dữ liệu cho Server, và khi client ngắt kết nối.  
     
   * Khi Client gửi dữ liệu lên, sự kiện sẽ được kích hoạt, ta convert mảng byte được gửi đi thành chuỗi String; sau đó Server gửi phản hồi về lại cho Client.  
     
   * Với các Client trong thời gian dài không gửi dữ liệu, ta sẽ ngắt kết nối.
   * Tại lớp **TcpCommunicationChannel** của thư viện Scs, ta thêm một thuộc tính **Timer** đến thời gian Client không gửi dữ liệu (mặc định là 10s). Khi có một Client kết nối tới, lớp này sẽ tạo ra Socket cho việc trao đổi dữ liệu của 2 thiết bị. Ta Start Timer ở đây và đăng ký sự kiện khi chạy đủ thời gian quy định.  
     
   * Phương thức xử lý cho sự kiện này sẽ gọi phương thức Disconnect để ngắt kết nối Client.  
     
   * Khi có dữ liệu được gửi đi, ta Stop Timer lại để chương trình xử lý. Khi xử lý xong, ta đếm lại từ đầu.  
     