**HỆ THỐNG ĐỊNH VỊ XE MÁY**

**BÁO CÁO TUẦN 2**

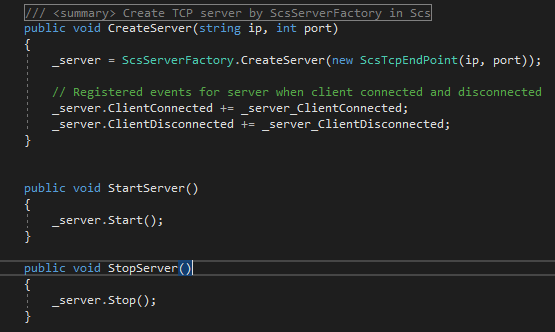
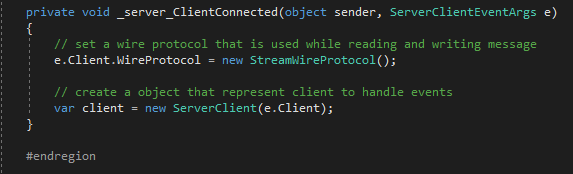
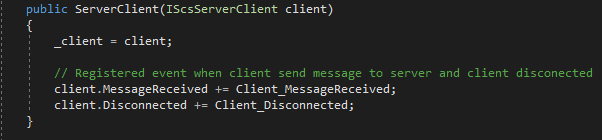
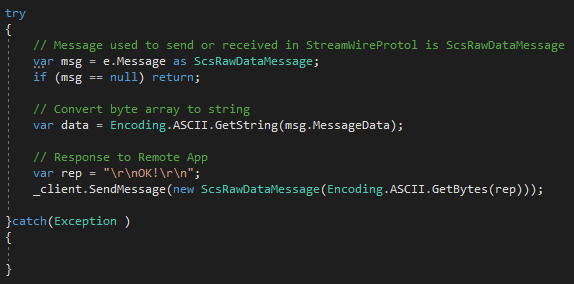
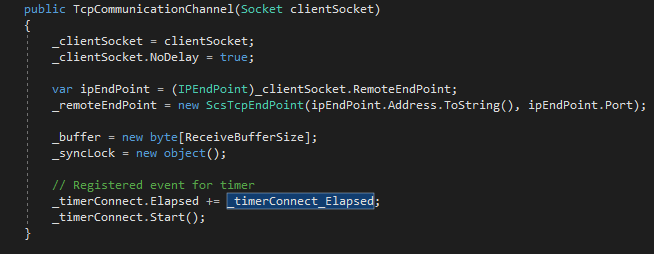
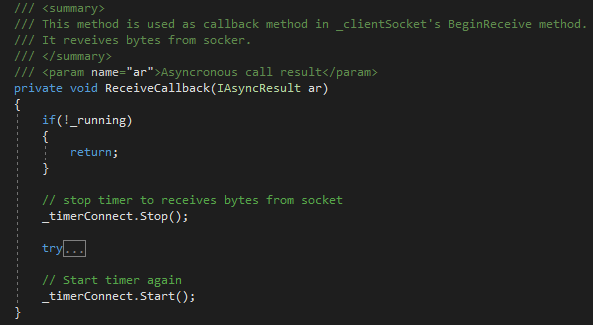
Mục lục

[***1.*** ***Công việc hoàn thành:*** 3](#_Toc4923684)

[**2.** ***Cấu trúc chương trình:*** 3](#_Toc4923685)

[***3.*** ***Mô tả hoạt động:*** 3](#_Toc4923686)

[**4.** ***Bug***: 6](#_Toc4923687)

1. ***Công việc hoàn thành:***
   * Tạo ra một Web API project, khi chạy tạo ra một TCP Server.
   * Đọc thông số cấu hình từ file xml.
   * 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:***
   * **Scs**: thư viện Scs.
   * **ConfigFile**: Dùng để cấu hình địa chỉ Ip và Port:  
     - **IconfigManager**: Interface có 2 method là read và write lần lượt dùng để đọc và viết file cấu hình.  
     - **IconfigObject**: method Fix dùng giá trị mặc định để cấu hình Ip và Port.  
     - **ConfigManager**: implement IconfigManager.
   * **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 chi tiết:***
   * Tạo ra đối tượng **TcpServer**, gọi methods đọc cấu hình từ file xml “Config.xml” trong thư mục bin.
   * Sau khi cấu hình, tạo Server từ Ip và Port có được. 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** là giao thức 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.  
     
   * 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 Socket được tạo 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.  
     
4. ***Bug***:
   * Nếu ban đầu không có file cấu hình, chương trình sẽ dùng thông số mặc định để cấu hình và ghi ra file cấu hình vào thư mục bin. Vì vậy ứng dụng sẽ bị Restart và khi tạo socket  
     