CHƯƠNG 6: ỨNG DỤNG GIÁM SÁT VÀ ĐIỀU KHIỂN

6.1. Thiết kế giao diện điều khiển bằng C#

6.1.1. Tổng quan giao diện chính



*Hình 6.1.1: Tổng quan về giao diện chính*

Connect: Vùng kết nối với PLC với máy tính thông qua C#.

* Connect: Kết nối với PLC với C#.
* Disconnect: Ngắt kết nối PLC với C#.

Control: Vùng thiết lập các tiền tố cho việc điều khiển robot.

* Servo: Bật/Tắt các module điều khiển AC Servo cho cánh tay robot.
* Reset error: Reset lại các lỗi xuất hiện trên module về trạng thái ban đầu.
* Set home: Gán vị trí hiện tại là gốc của robot.
* Go home: Điều khiển robot về vị trí gốc.

Jogging: Vùng điều khiển việc chạy Jog robot.

* Set speed: Thiết lập tốc độ cho các AC Servo với giá trị bên trái.
* Joint []: Chọn khớp cần chạy Jog.
* Forward: Chạy theo chiều thuận.
* Backward: Chạy theo chiều nghịch.

Display: Vùng hiển thị vị trí đầu tay gắp của Robot.

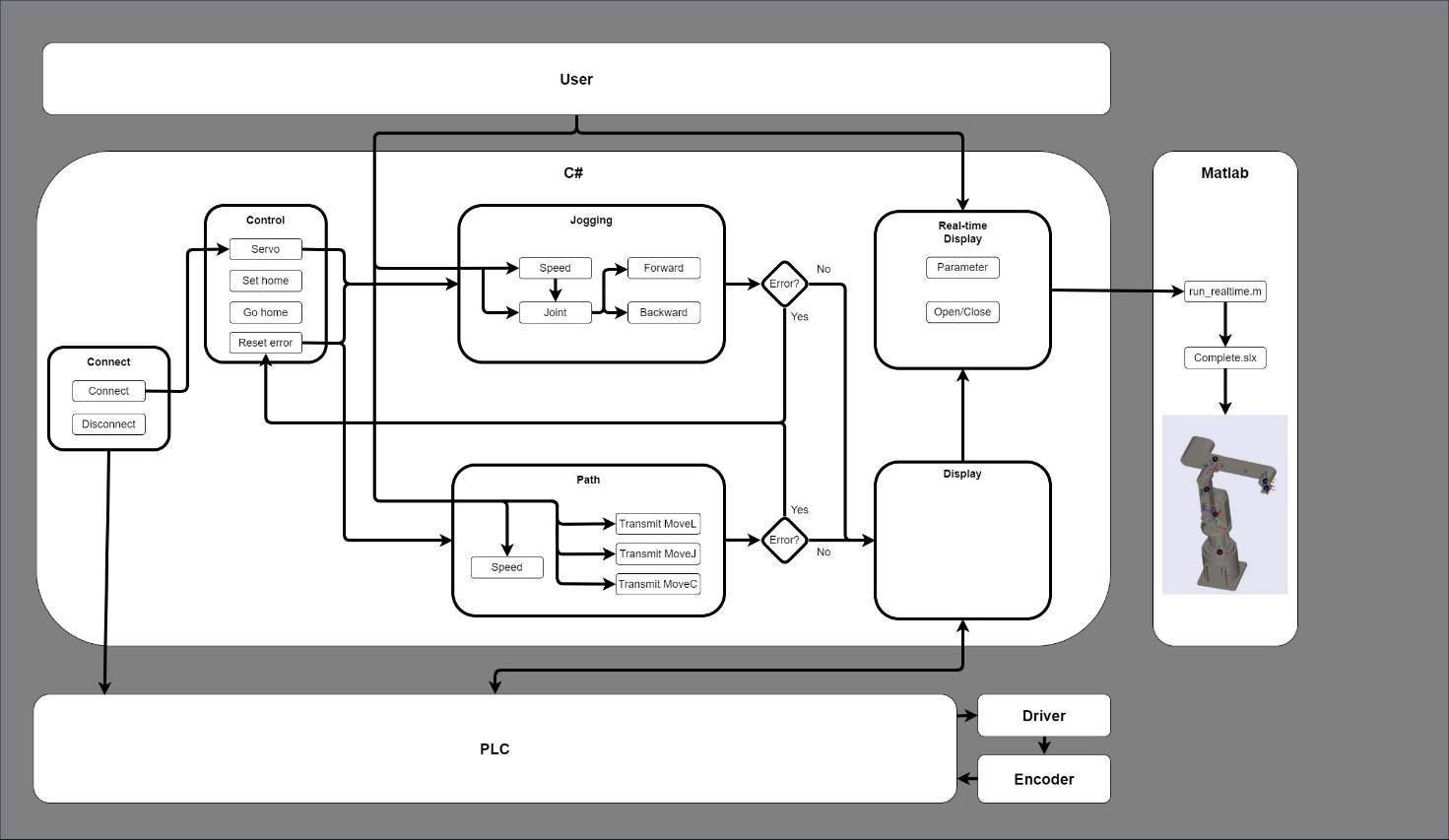
* X, Y, Z, Pitch, Roll: Hiển thị vị trí của đầu tay gắp.
* t1, t2, t3, t4, t5: Các giá trị góc nhập vào để điều khiển robot dựa theo động học thuận.
* Start: Khởi động timer của C#.
* Stop: Tắt timer của C#.

Real-time Display: Vùng kết nối giữ robot 3D và robot thực tế.

* Comport: Lựa chọn cổng COM.
* Baudrate: Lựa chọn tốc độ truyền.
* Data bits: Số bits data truyền đi.
* Stop bits: Lựa chọn bit cuối.
* Parity bits: Lựa chọn parity bits.
* Start Simulink: Kết nối với robot 3D trong Matlab.
* Close: Đóng kết nối.
* Start record: Ghi lại giá trị cái góc vào file csv.
* Stop record: Ngừng ghi.
* OPEN/CLOSE: Bật/tắt cổng COM.

Path: Vùng điều khiển robot.

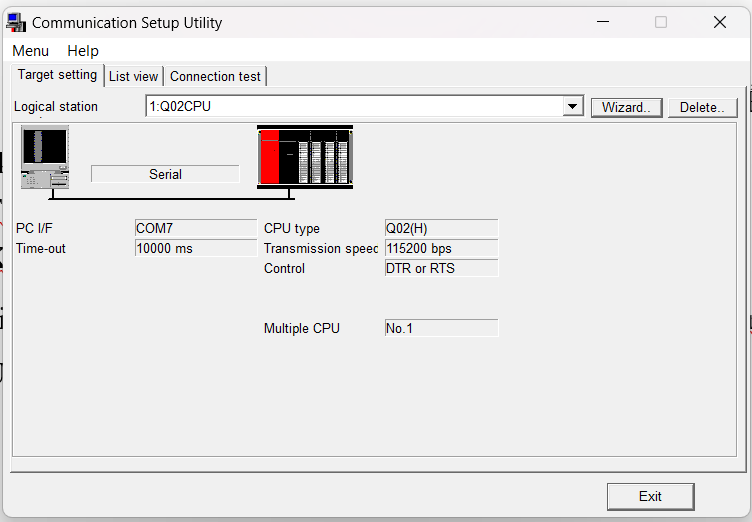
* Run: Chạy Robot thực tế.
* Trasmit MoveJ: Gửi tín hiệu điều khiển theo MoveJ vào vùng nhớ PLC.
* Trasmit MoveC: Gửi tín hiệu điều khiển theo MoveC vào vùng nhớ PLC.
* Trasmit MoveL: Gửi tín hiệu điều khiển theo MoveL vào vùng nhớ PLC.
* Set const speed: Gửi giá trị vận tốc vào vùng nhớ PLC.



*Hình 6.1.1: Sơ đồ nguyên lí hoạt động của hệ thống*

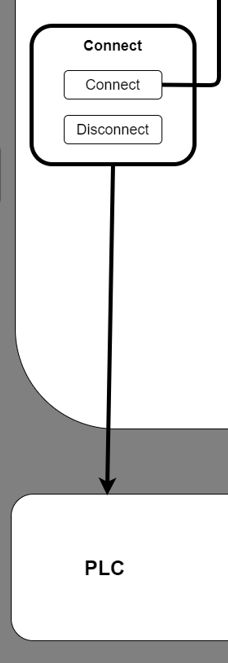
6.1.2. Kết nối với PLC từ máy tính

PLC Mitsubishi hỗ trợ thư viện ActUtlTypeLib thông qua phần mềm Communication Setup Utility để có thể kết nối được giữa PLC và C# thông qua serial port RS232.



*Hình 6.1.2a: Giao diện phần mềm Communication Setup Utility*

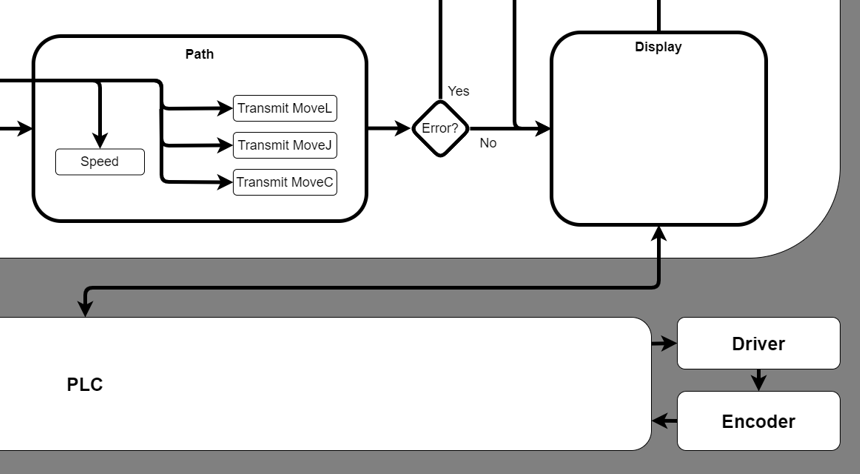
Sau khi setup trên phần mềm, ta khai báo cách kết nối trên C# bằng một số câu lệnh:



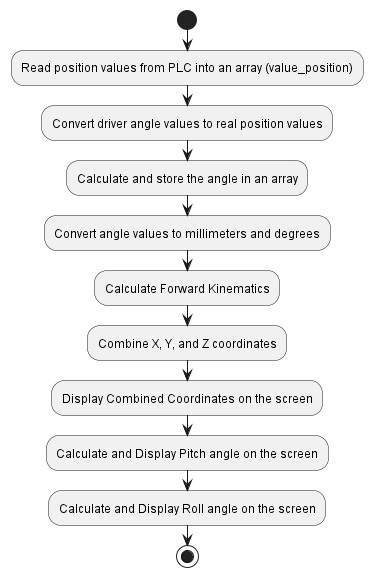
*Hình 6.1.2b: Sơ đồ nguyên lí kết nối PLC với Máy tính thông qua C#.*

6.1.3. Đọc giá trị góc trả về từ PLC

Phầm mềm Communication Setup Utility hỗ trợ việc đọc giá trị trên thanh ghi 16 bits của PLC giúp chúng ta có thể tương tác cũng như dựa vào đó, tính toán và tương tác với cánh tay Robot thực tế.



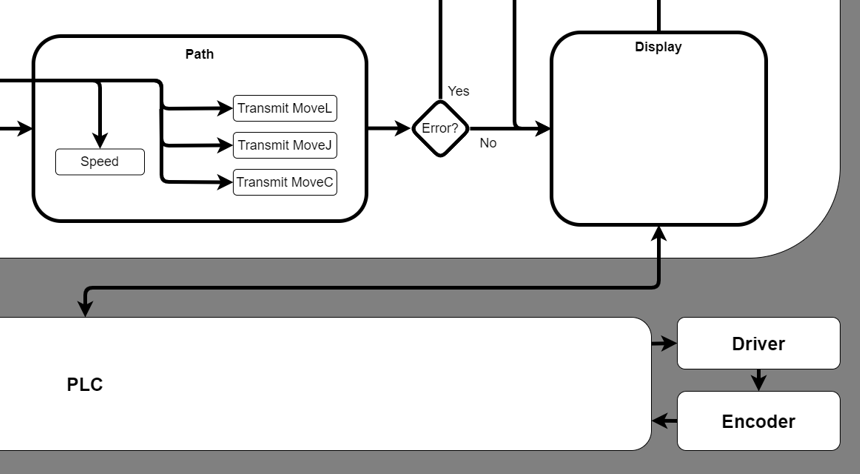
*Hình 6.1.3a: Sơ đồ nguyên lí đọc giá trị góc trả về từ PLC*



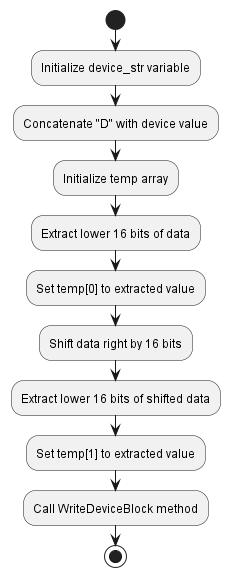
*Hình 6.1.3b: Lưu đồ giải thuật đọc giá trị góc trả về từ PLC để tính động học thuận*

6.1.4. Gửi câu lệnh điều khiển xuống PLC

Tương tự, việc gửi dữ liệu lên PLC cũng có thể thực hiện.



*Hình 6.1.3b: Sơ đồ nguyên lí gửi tín hiệu điều khiển xuống PLC*

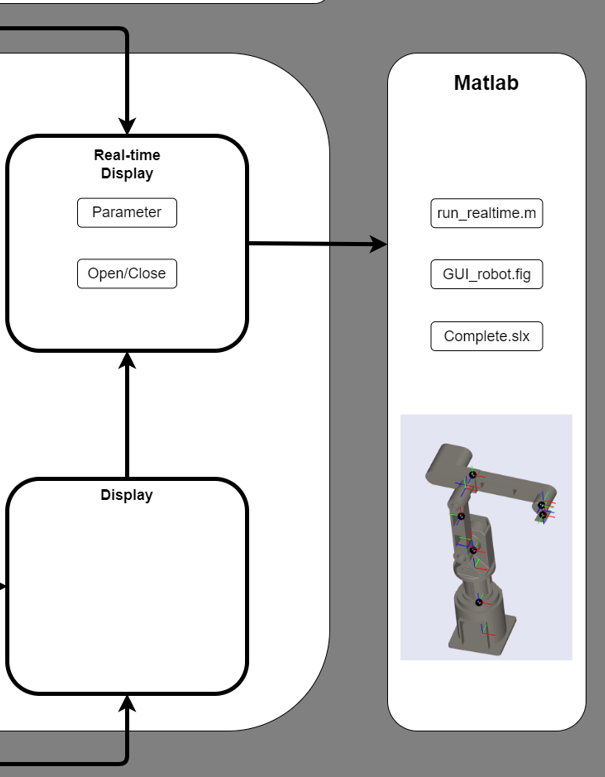


*Hình 6.1.3c: Lưu đồ giải thuật cho việc gửi giá trị xuống vùng nhớ PLC*

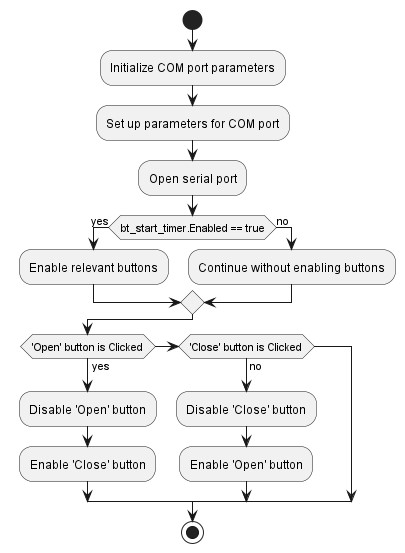
6.1.5. Kết nối với MATLAB để thực hiện xử lí dữ liệu góc

Để tăng hiệu quả trong việc điều khiển, việc thể hiện robot thực tế lên phần mềm máy tính là cần thiết. Do vậy, Matlab là một công cụ mạnh mẽ để có thể mô phỏng được cánh tay robot ngoài thực tế.

Bằng việc kết nối giữa Matlab và C#, ta có thể dễ dàng trao đổi dữ liệu giữa 2 phần mềm một cách đơn giản thông qua cổng COM ảo có thể được thiết đặt bởi các phần mềm hiện có trên thị trường như Configure Virtual Serial Port Driver, Virtual Serial Port Emulator…



*Hình 6.1.3c: Sơ đồ nguyên lí gửi dữ liệu từ C# qua Matlab*



*Hình 6.1.3d: Lưu đồ giải thuật kết nối C# với Matlab thông qua cổng COM*