**DATABASE SERVER (TF6420)**

**WITH SQLITE DOCUMENT**

1. **Giới thiệu SQLite**

SQLite là một thư viện phần mềm mà triển khai một SQL Database Engine, không cần máy chủ, không cần cấu hình, khép kín và nhỏ gọn. Nó là một cơ sở dữ liệu, không cần cấu hình, có nghĩa là giống như các cơ sở dữ liệu khác mà bạn không cần phải cấu hình nó trong hệ thống của mình.

SQLite engine không phải là một quy trình độc lập (standalone process) như các cơ sở dữ liệu khác, bạn có thể liên kết nó một cách tĩnh hoặc động tùy theo yêu cầu của bạn với ứng dụng của bạn. SQLite truy cập trực tiếp các file lưu trữ (storage files) của nó.

SQLite được sử dụng rộng rãi vì:

* SQLite không yêu cầu một quy trình hoặc hệ thống máy chủ riêng biệt để hoạt động.
* SQLite không cần cấu hình, có nghĩa là không cần thiết lập hoặc quản trị.
* Một cơ sở dữ liệu SQLite hoàn chỉnh được lưu trữ trong một file disk đa nền tảng (cross-platform disk file).
* SQLite rất nhỏ và trọng lượng nhẹ, dưới 400KiB được cấu hình đầy đủ hoặc dưới 250KiB với các tính năng tùy chọn bị bỏ qua.
* SQLite là khép kín (self-contained), có nghĩa là không có phụ thuộc bên ngoài.
* Các transaction trong SQLite hoàn toàn tuân thủ ACID, cho phép truy cập an toàn từ nhiều tiến trình (process) hoặc luồng (thread).
* SQLite hỗ trợ hầu hết các tính năng ngôn ngữ truy vấn (query language) được tìm thấy trong tiêu chuẩn SQL92 (SQL2).
* SQLite được viết bằng ANSI-C và cung cấp API đơn giản và dễ sử dụng.
* SQLite có sẵn trên UNIX (Linux, Mac OS-X, Android, iOS) và Windows (Win32, WinCE, WinRT).

1. **Giới thiệu về Database Server (TF6420)**

TwinCAT Database Server cho việc trao đổi dữ liệu giữa TwinCAT với nhiều database khác nhau. Cho những ứng dụng đơn giản chúng ta có thể dùng thông qua Configurator mà không cần phải sử dụng code. Cho những chương trình có yêu cầu phức tạp hơn thì Database Server cung cấp thư viện để thực hiện một cách linh hoạt hơn. Câu lệnh SQL như là Insert, Select có thể thực hiện trực tiếp trên PLC

TwinCAT Database Server hỗ trợ nhiều hệ thống database khác nhau như: MS SQL, MS SQL Compact, MS Access, MySQL, PostgreSQL, DB2, Oracle, Interbase, Firebird, ASCII, XML file, và bây giờ Database Server cũng hỗ trợ cả NoSQL database (MongoDB)

Tổng cộng có 4 chế độ cấu hình cho Database Server:

* Configure mode: Kết nối database cho những ứng dụng đơn giản và cấu hình dựa trên giao diện người dùng không cần code.
* PLC Expert mode: một phương án lập trình cho những nhà lập trình PLC tiện lợi nhất.
* SQL Expert mode: phương án lập trình cho những nhà lập trình PLC và hiểu biết nhiều về SQL.
* NoSQL Expert mode: phương án lập trình cho nhà lập trình PLC và hiều biết nhiều về database NoSQL.

Trong tài liệu này sẽ trình bày chi tiết về 2 chế độ của TF6420 là Configure mode và PLC Expert mode và 2 chế độ này là 2 chế độ tối ưu khi triển khai cùng với SQLite.

1. **Cài đặt TF6420 trên IPC**

Cài đặt thư viện TF6420 trên Laptop Engineering trước.

Đối với IPC chạy Windows 10/7 thì có thể cài TF6420 bằng cách bình thường bằng file cài đặt tải từ trang Beckhoff

Đối với IPC chạy Windows CE thì cần sử dụng file có phần mở rộng \*.CAB.

Có thể truyền file \*.CAB thông qua các cách thức sau:

* Thông qua network shares.
* Thông qua FTP server.
* Thông qua ActiveSync.
* Thông qua CF/SD cards.

Phương pháp đã sử dụng và thành công là FTP server. Sử dụng phần mềm WinSCP.

Cách sử dụng WinSCP với chế độ FTP xem thêm trên Internet.

File CAB chứa trong thư mục **C:/TwinCAT/Functions/TF6420-Database-Server/CE-ARM**

Sau khi truyền file qua IPC thì tiến hành cài đặt Database Server một cách bình thường bằng cách Double Click vào file CAB vừa copy vào.

Lưu ý để khởi động trình File Explorer trong Win CE bằng cách vào Start -> Run gõ “Explorer”

1. **Database Server với SQLite với Configure mode**

Chế độ Configure mode này là một chế độ đơn giản không cần code PLC và SQL nhiều. Tuy nhiên đối với Database phức tạp thì không nên sử dụng mode này để ghi Database.

Ở chế độ Configure mode, Database cho phép ta cấu hình kết nối với SQLite Database, tùy chỉnh database (INSERT, DELETE, CREATE TABLE) thông qua SQL Qurey Editor, tạo các bảng AutoLogGroup.

1. Các bước thực hiện

**Project: DatabaseServer\_Configure\_01**

Lưu ý: Sau từng bước Config các thành của Project Connectivity cần phải Activate Configuration.

Tạo Project TwinCAT Connectivity.

Bên trong Project TwinCAT Connectivity Tạo Project TwinCAT Database Server Project.

Bên trong Project TwinCAT Database Server Project Add New Database.

Bên trong Database vừa tạo, tạo New AutologGroup. (Right-Click vào Database vừa tạo chọn Add New AutologGroup)

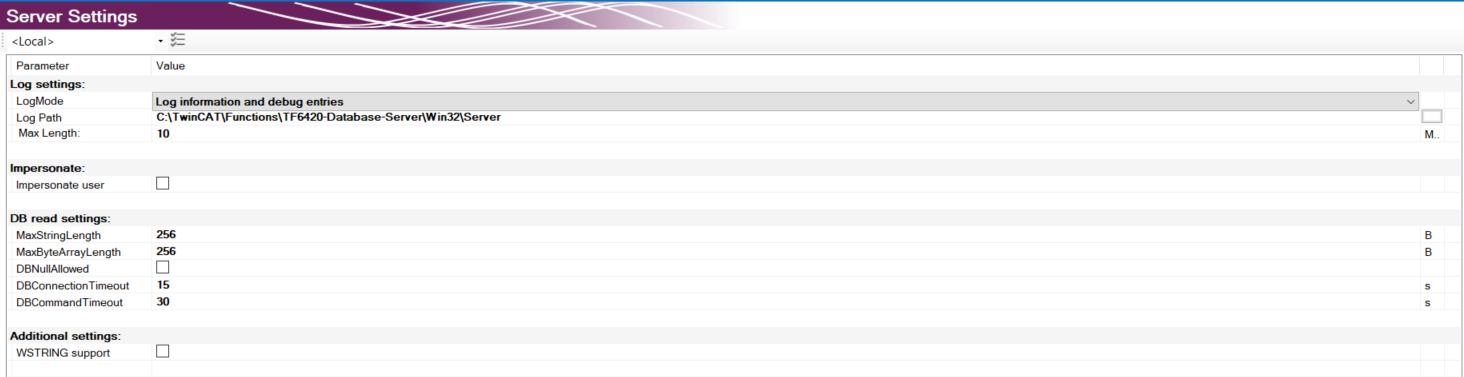
**Config Project DatabaseServer:**

* Log mode
* Log Path
* Vị trí truy cập Database Server: Local/Target: 

Log ở đây sẽ ghi log lại những sự kiện, ngoại lệ trên file/thư mục đã trỏ đến. Nếu không quá quan trọng thì có thể để giống như tùy chọn mặt định của nó

Vị trí truy cập DatabaseServer rất quan trọng sẽ được trình bày ở mục Config kết nối với IPC thực.

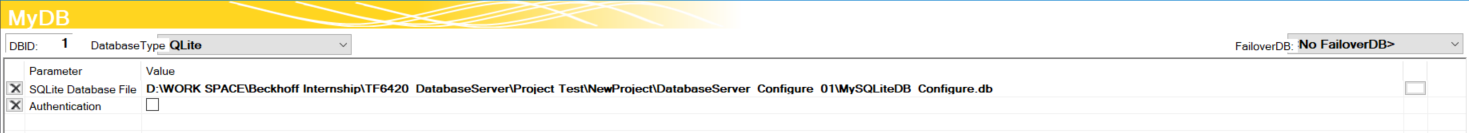
Sau khi đã điền các thông số:



**Config Database:**

* DBID: là chỉ số trỏ đến Database này, DBID không trùng giữa các Database
* DatabaseType: Chọn tương ứng với Database sử dụng. Ở đây sử dụng SQLite (SQLite hỗ trợ WinCE)
* FailoverDB: đây là một tính năng mở rộng của Database Server, không bắt buộc phải thiết lặp. Ở đây sẽ trỏ đến một database khác. Database này sẽ được gọi đến khi lỗi xảy ra ở chế độ Configure mode. Trong trường hợp mất kết nối với mạng thì chức này có thể tự động bật lên và đảm bảo dữ liệu sẽ được lưu trữ mà không bị mất.
* SQLite Database File: là đường link dẫn đến file database SQLite (\*.db)
* Authentication: liên quán đến bảo mật. Để đơn gian hóa trong việc triển khai thì chúng ta có thể không check vào mục này.

Sau khi đã điền các thông số:



Sau khi đã điền đầy đủ các thông số như trên nhấn nút kiểm tra CHECK ở bến dưới. Nếu kết quả báo về là Succeded thì chuyển sang bước tiếp theo.

**Config AutologGroup:**

* AutoLogGrpID: chỉ số trỏ đến AutologGrp
* StartUp: sẽ xác định khi nào mà bảng sẽ tự động ghi giá trị. Có hai chế độ là AutoStart và Manual. AutoStart thì dữ liệu sẽ ghi tự động khi IPC run. Manual thì dữ liệu sẽ bắt đầu ghi vào bảng khi bằng câu lệnh ở chương trình PLC hoặc từ thao tác điều khiển tay từ màn hình Configurator.
* Direction: chọn flow truyền của dữ liệu

DeviceAsSource: Xem Device là nguồn cấp dữ liệu lấy dữ liệu từ Device lên ghi vào bảng

DeviceAsTarget: Xem Device là đối tượng cần ghi dữ liệu xuống.

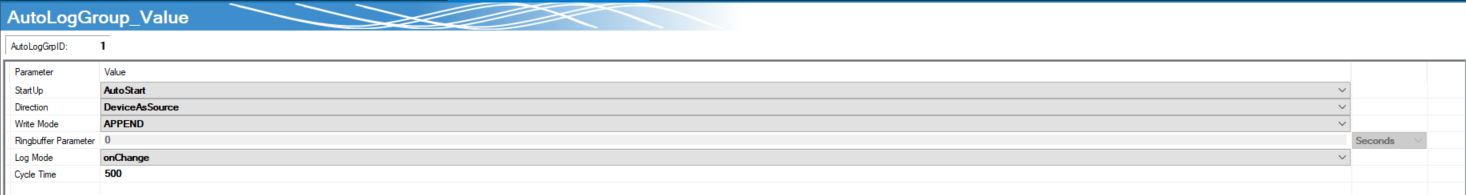
* Write mode: chế độ ghi. APPEND là chế độ ghi theo line-by-line.
* LogMode: chế độ log.

Cyclic: Log theo chu kì cố định

onChange: Log khi mà value của biến thay đổi

* Cycle Time: thời gian sau khi mà biến được ghi vào bảng.

Điền đầy đủ các thông số như sau:



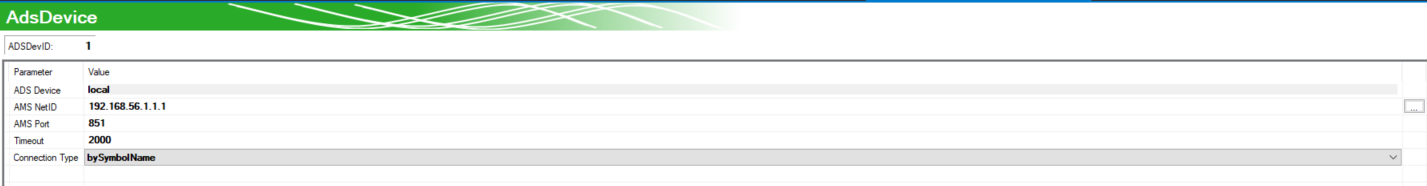
**Config AdsDevice trong AutologGrp**

* ADSDevID: chỉ số trỏ đến.
* AMS NetID: trỏ đến thiết bị ADS muốn kết nối (ở đây trỏ đến Local).
* AMS port: 851.
* Time out: Thời gian timeout
* Connection Type:

bySymbolName: kết nối thiết lập thông qua tên biến

byIndexGroup: kết nối thiết lập thông qua memory index

Sau khi đã điền đầy đủ các thông tin:

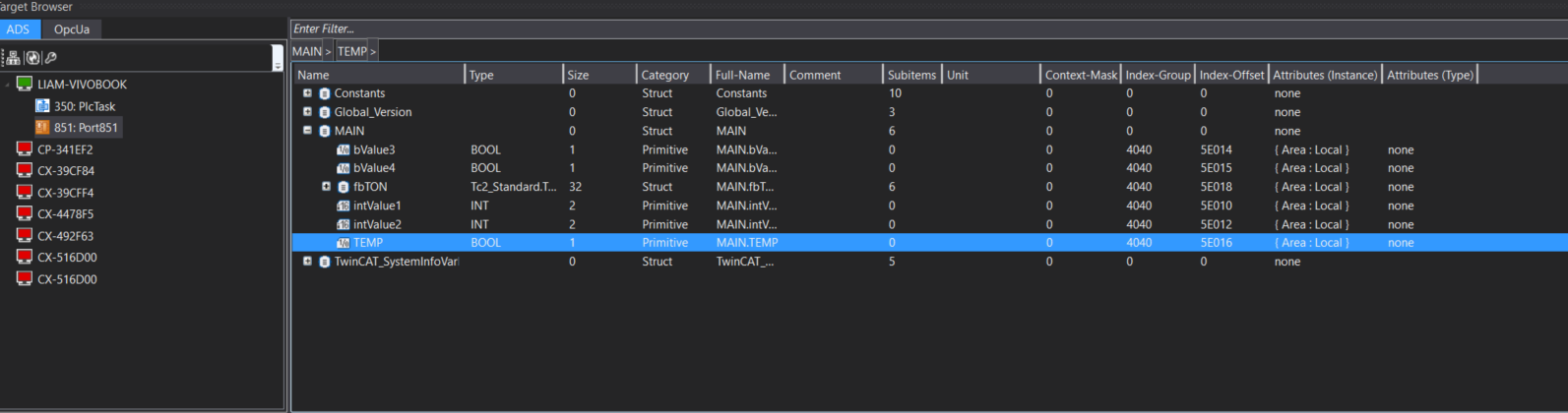


**Configuring symbols:**

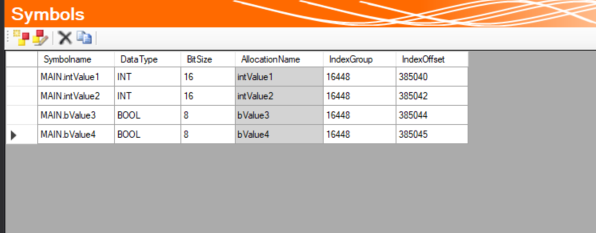
Những Symbol được config là những Symbol cần được đọc hoặc ghi vào database phụ thuộc vào việc set ADS Sevice là source hay target. TwinCAT Target browser có thể được sử dụng để thuận tiện truy cập hơn. Sử dụng thao tác drag and drop để thiết lập Symbol.

Để mở Target Browser vào Menu TwinCAT trên thanh Menu của Visual, Vào Target Browser, chọn taraget Browser.

Cửa sổ Target Browser hiện lên, chọn vào máy tính Local, chọn Port 851.

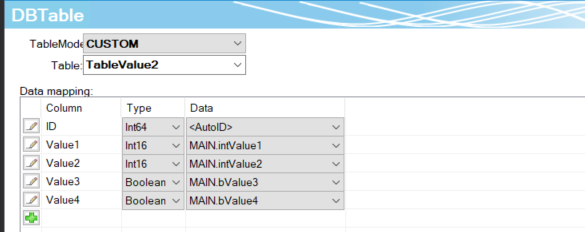


Sau đó kéo thả các biến cần Config vào Symbols của AutologGrp.

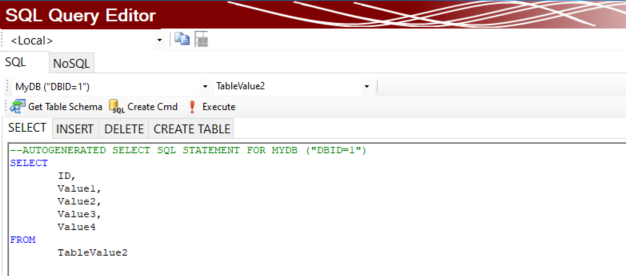


**Configuring DBTable trong AutologGrp:**

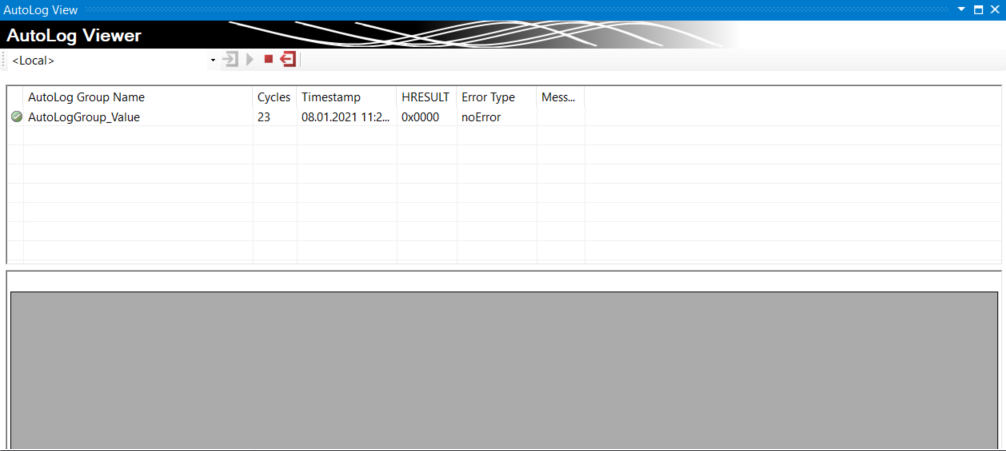
TableMode: có thể lựa chọn STANDARD là kiểu table có sẵn, hoặc chọn CUSTOM là kiểu bảng mà người dùng tự tạo.



Có thể sữ dụng SQL Query Editor để đọc ghi, thực hiện câu truy vấn với database. Cách truy cập: Menu TwinCAT → Database Server → SQL Query Editor.

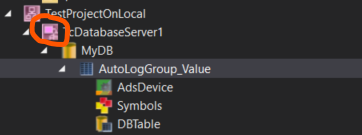


Kiểm tra AutoLogGrp có hoạt động bình thường hay không bằng AutoLog View:



Để kiểm tra ở cửa sổ này cần Log in vào, sau đó nhất Start trên cửa sổ.

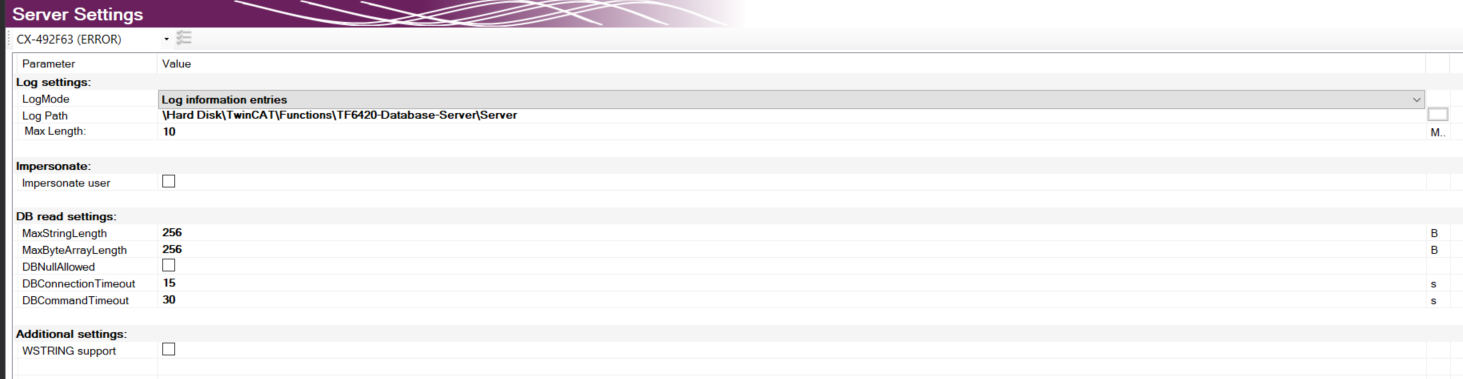
**Lưu ý** sau khi đã kiểm tra bảng AutologGrp thì cần phải Stop và Logout ra thì mới có thể chỉnh sửa những phần khác của project. Khi Database Server đang ở chế độ Autolog thì kí hiệu đèn trên cây thư mục sẽ màu tím:



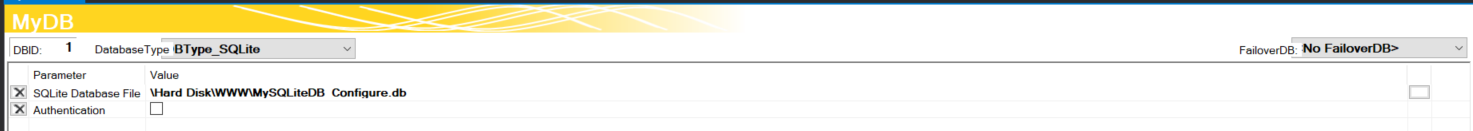
Sau khi hoành thành các bước trên thì khi cho IPC Run và giá trị các biến thay đổi thì sẽ log vào TableValue2 của Database SQLite lúc trước chúng ta vừa Config

1. Config khi kết nối IPC thực

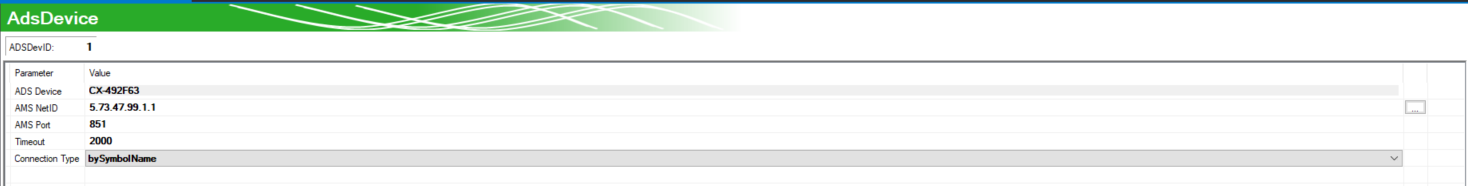
Config lại Project Database Server:



Config lại phần Database ở đường Link dẫn đến file SQLite trên IPC:



Config phần ADS Device



1. **Database Server – SQLite với PLC Expert mode**

Trong chế độ PLC Expert chỉ có phần configuration (thiết đặt ban đầu) là thực hiện bằng configurator trên Visual. Những chức năng khác như đọc ghi, truy vấn sẽ được thực thi bằng code PLC.

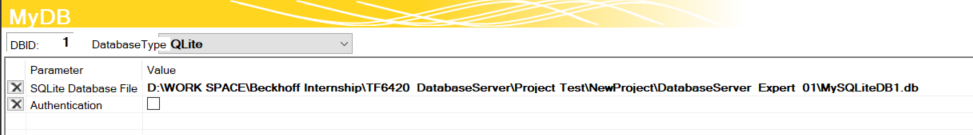
1. Các bước thực hiện

**Project: DatabaseServer\_Expert\_01**

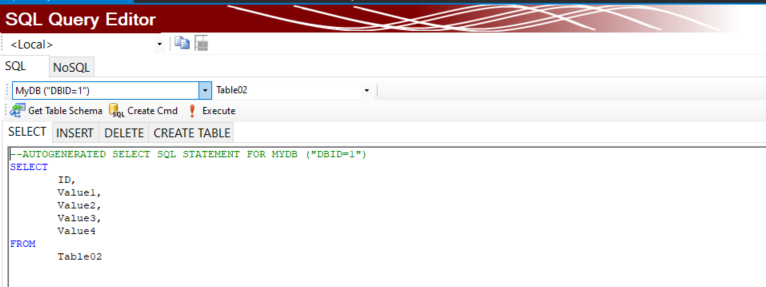
Thực hiện các bước khởi tạo Project như mode Configuration.

Tuy nhiên chúng ta không tạo AutologGrp. Chỉ dừng lại ở bước New Database.





**Lưu ý** để tránh bị lỗi do không hợp lệ kiểu dữ liệu chúng ta nên tạo các bảng bằng công cụ SQL Query Editor có sẵn trong Menu TwinCAT trong Visual.

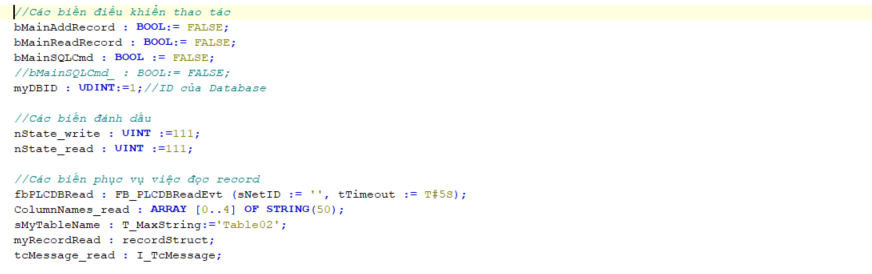


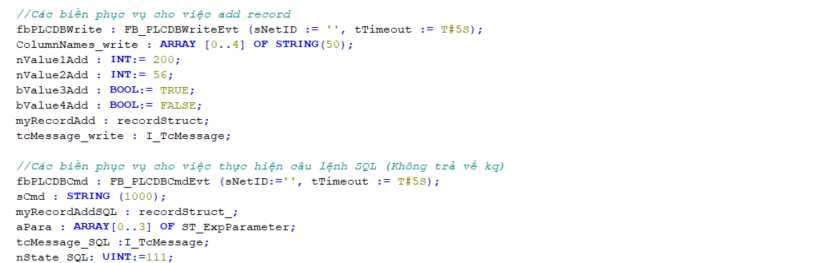
Dùng chức năng CREATE TABLE

**Code PLC:**

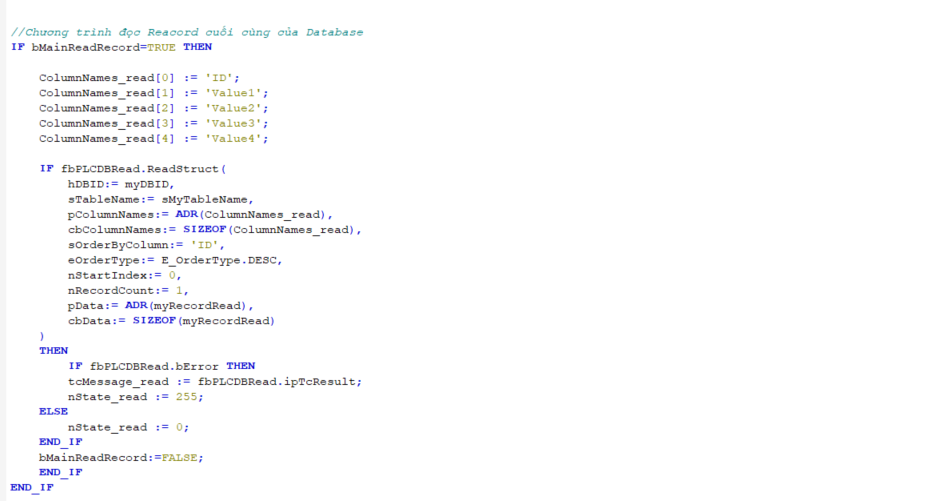
Khai báo các biến:

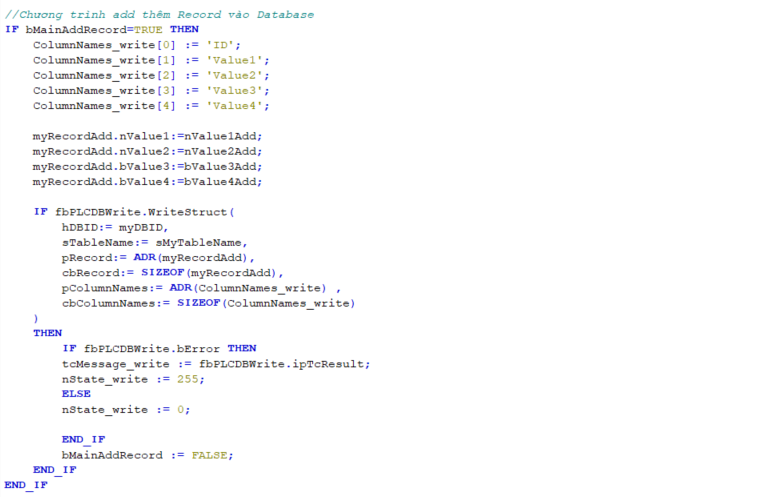
*Ý nghĩa của các biến được chú thích tương ứng theo các dòng code.*

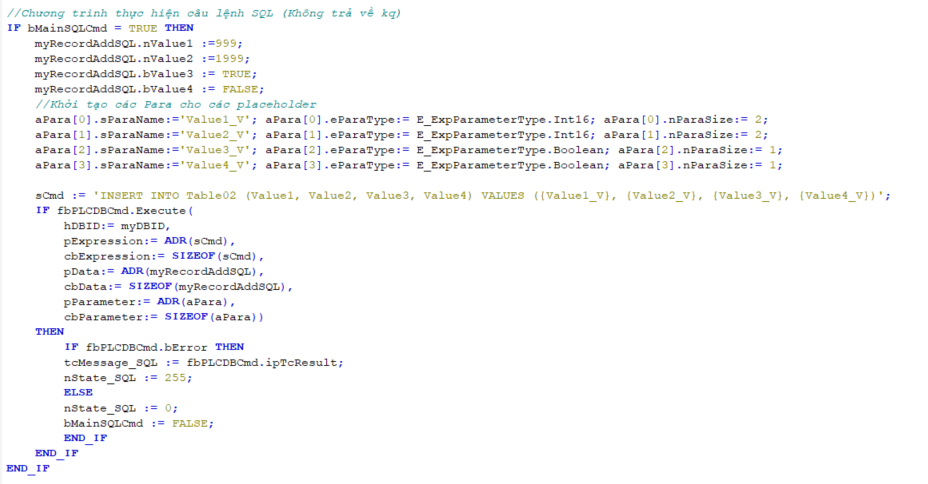




Chương trình MAIN của PLC:







**Giải thích chương trình:**

**Đọc record cuối cùng:**

Sử dụng Method của FB: FB\_PLCDBRead.ReadStruct()

Các đối số cần truyền vào là: DBID, Tên bảng, địa chỉ mảng chứa tên các cột, kích thước mảng tên các cột, Order, OrderType, Index bắt đầu, Số lượng record đọc, địa chỉ chứa data trả về, kích thước data trả về.

Kèm với sử dụng Method này cần phải có phương pháp xử lý những lỗi có thể xảy ra.

***Lưu ý*** khi Method ReadStruct được gọi thì hàm này sẽ thực thi và trả về kết quả là true khi thực hiện xong dù có gặp lỗi hay không.

**Thêm Record vào Database:**

Các đối số cần truyền vào: DBID, Tên bảng, địa chỉ mảng chứa Record cần đưa vào, kích thước, địa chỉ mảng chứa các tên của cột, kích thước.

Kèm với sử dụng Method này cần phải có phương pháp xử lý những lỗi có thể xảy ra.

**Lưu ý** khi Method này được gọi thì hàm này sẽ thực thi và trả về kết quả là true khi thực hiện xong dù có gặp lỗi hay không.

**Thực thi câu lệnh SQL:**

**Lưu ý** thực thi câu lệnh SQL bằng FB\_PLCDBCmd.Execute() còn một số hạn chế nhưng chỉ thực hiện được những câu SQL có cấu trúc bảng trả về xác định trước, không thực hiện được các câu lệnh SQL tạo bảng, thêm cột, … Những chức năng chuyên về sử dụng câu lệnh SQL có thể sử dụng SQL Expert Mode.

Các thành phần cần chuẩn bị: câu SQL, mảng các Para cho placeHolder trong câu SQL

Ở đây câu SQL là câu lệnh Insert

'INSERT INTO Table02 (Value1, Value2, Value3, Value4) VALUES ({Value1\_V}, {Value2\_V}, {Value3\_V}, {Value4\_V})';

Các Value trong dấu { } là placeholder.

1. Config trên IPC thực

Config tương tự trong mode Configuration

1. Chú thích

Folder: DatabaseServer\_Expert\_01 chứa Project PLC Expert Mode

Folder: DatabaseServer\_Configure\_01 chứa Project PLC Configure mode

Folder: Test On CX chứa 2 Project config chạy trên IPC

Khi chạy Project trên máy tính khác cần thay đổi đường Link dẫn đến file Database (file Datanase có kèm theo trong Folder).