🔐 Thiết lập phần cứng

Các thiết bị cần có:

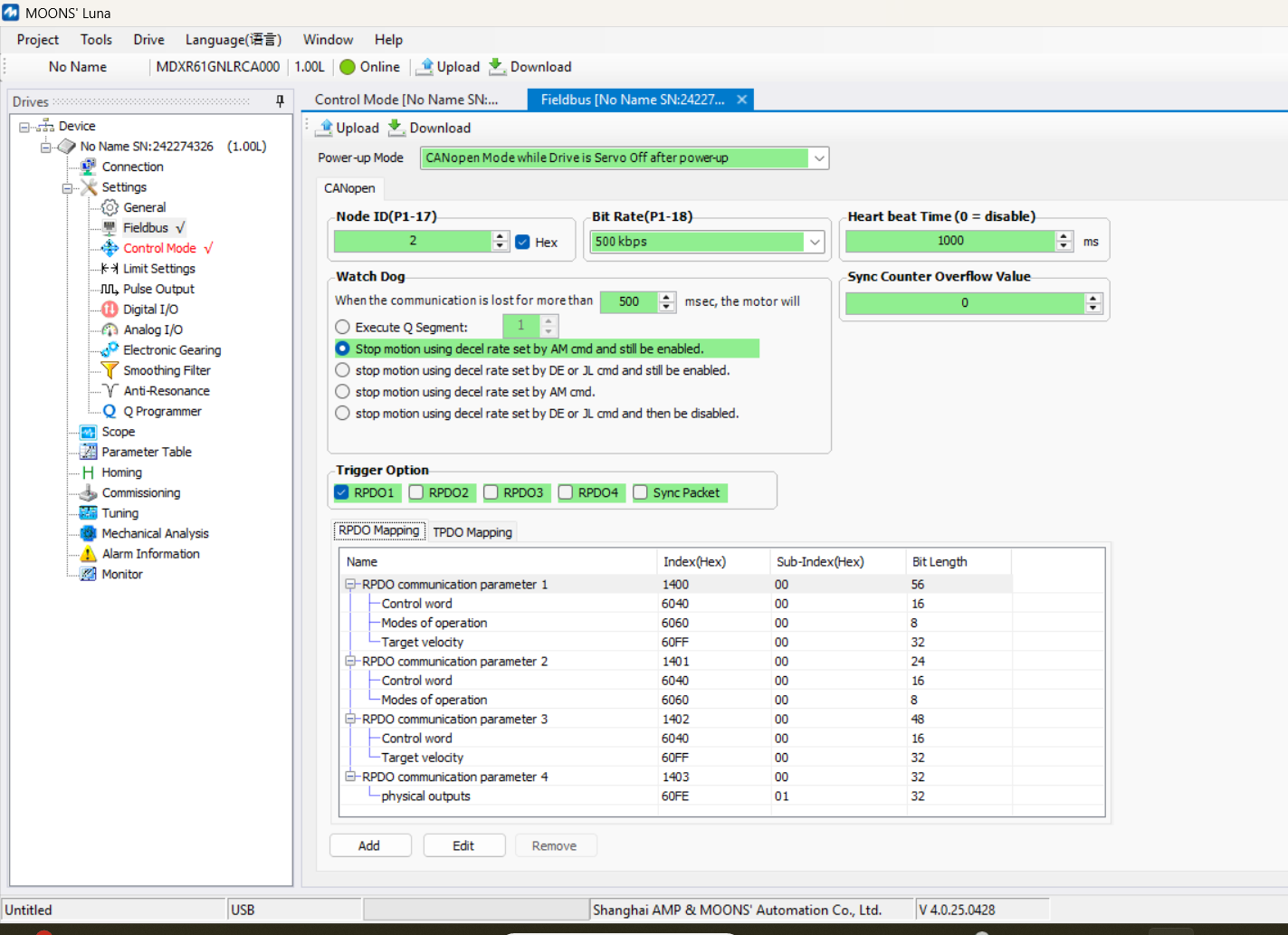
* PLC CPU S7-1200 G2 model 1214C DC/DC/DC
* Module I/O ET200SP IM 155 - 6PN/2 HF version 4.2.4
* Module CM 1xCAN ST version 1.1.0 (Base Unit màu trắng)
* Server module version 1.1.2
* MDX+ Integrated Servo
* Điện trở 120 ohm

Cách thức đấu nối:

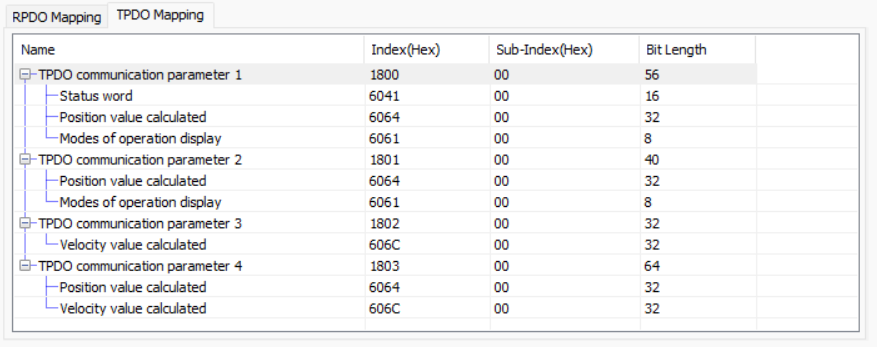
* Cấp nguồn 24VDC cho cả module CPU S7-1200 G, module ET 200SP, Base Unit của module CM CAN
* Cắm dây Ethernet từ chân P1 của S7-1200 G2 với chân P1R-X1 của ET 200SP
* 1 đầu Ethernet từ chân P2 cắm vào laptop để nạp code
* Tại Base Unit của module CM CAN, nối chân 1 CAN\_H nối với chân Can\_H của động cơ Servo, chân 2 Can\_L nối với chân 4 thông qua điện trở 120 ohm, từ chân 4 nối với chân Can\_L của Servo
* Cấp nguồn 24V cho chân AUX+, AUX-, V+, V-
* Tại cổng COM, nối chân GND với chân GND của hệ thống

🧩 Thiết lập cho file EDS

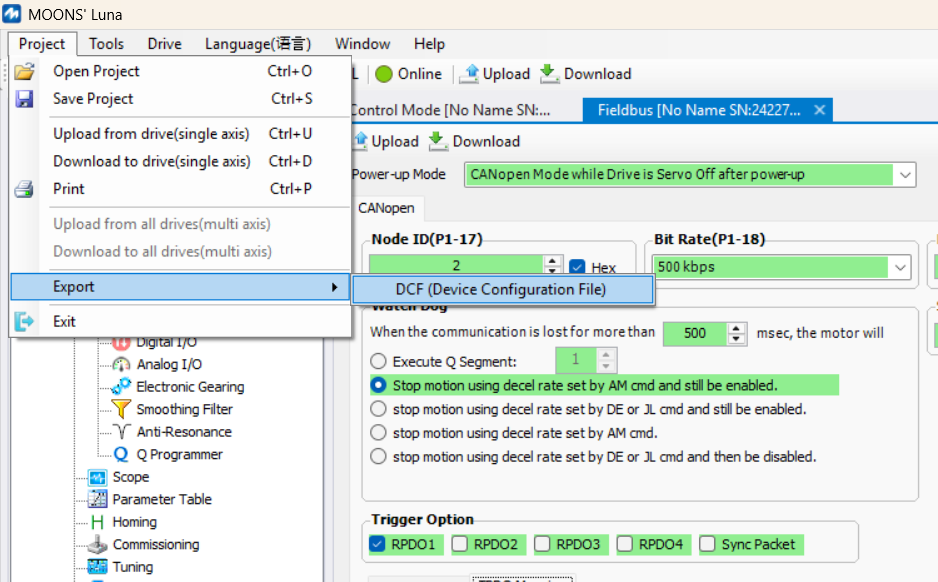
* Mở App Luna lên và mapping dữ liệu theo ý mình mong muốn



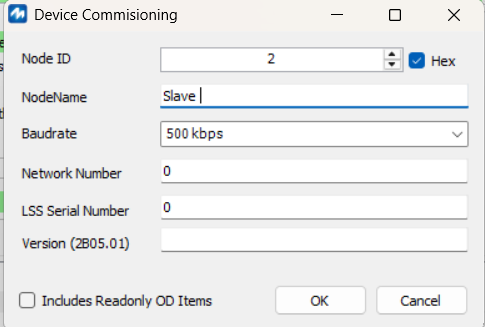
* Ví dụ với trường hợp đang xét điều khiển dạng Target Velocity. Bit Length tối đa của mỗi PDO là 8 byte tương đương 64 bit, vì vậy ta mapping cả 3 dữ liệu cần sử dụng vào trong RPDO1



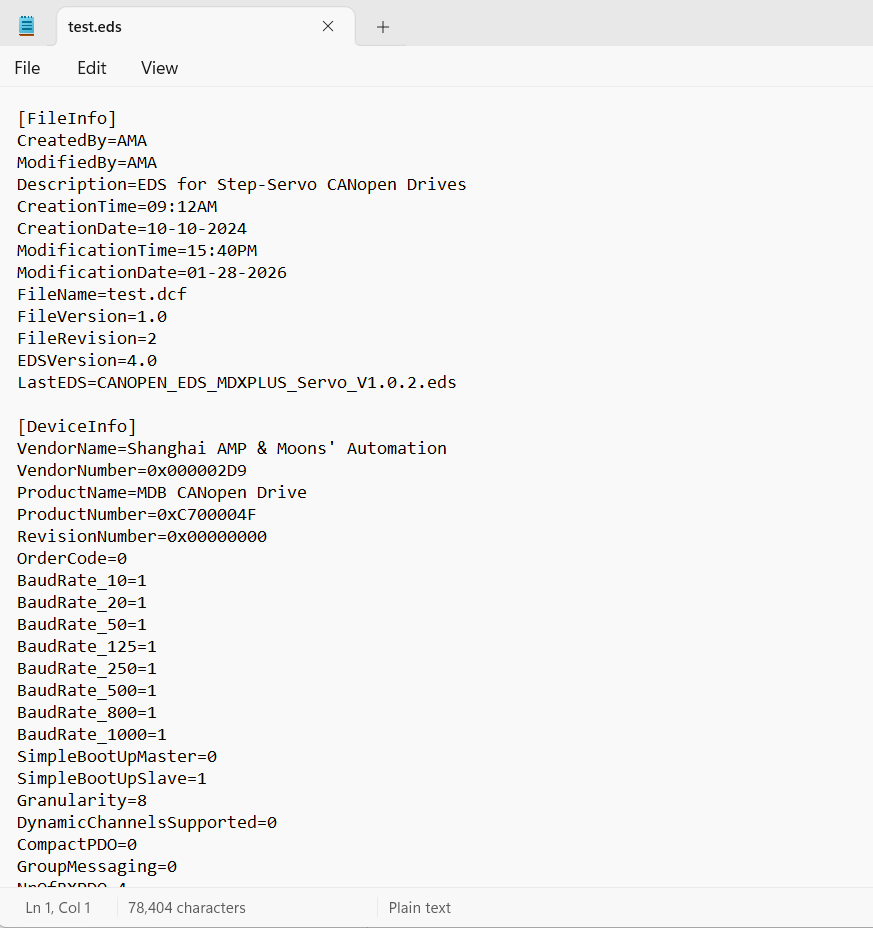
* Tương tự với việc mapping TPDO, 3 giá trị cần sử dụng có độ dài dưới 64 bit nên thỏa mãn yêu cầu độ dài tối đa của 1 PDO
* Lưu ý các thông số cần chú ý bao gồm Node ID là 2, Baut rate là 500kps, Heart beat là 1000ms
* Sau khi mapping các thông số xong, ta nhấn Download để nó lưu dữ liệu xuống động cơ. Sau đó chọn Project -> Export -> DCF (Device Configuration File)



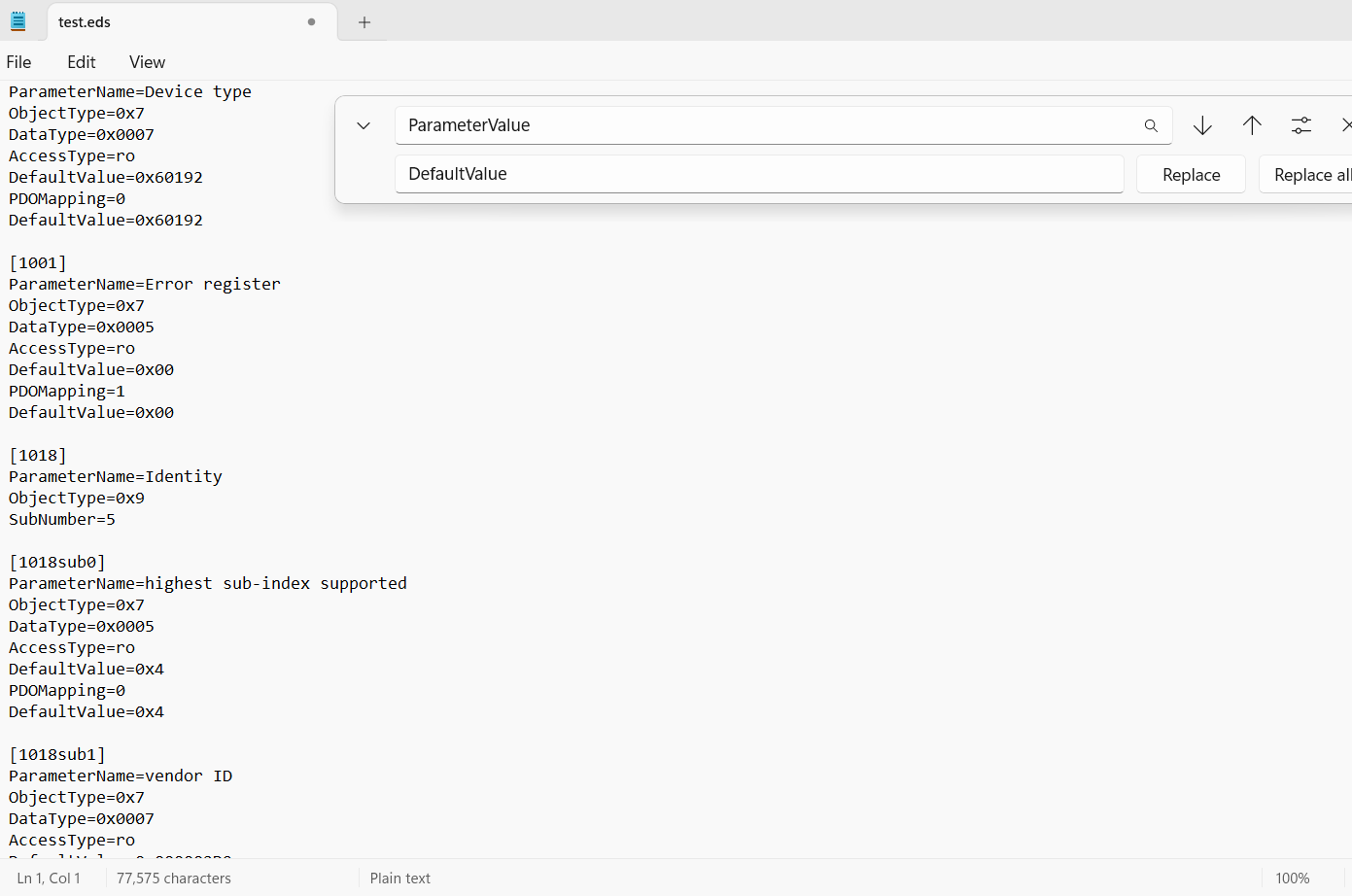
* Sau đó nhập NodeName mình mong muốn rồi nhấn OK, có thể nhấn OK luôn cũng được



* Sau khi tạo được file sẽ có dạng .dcf, t cần rename nó để thành dạng eds, rồi mở file đó trên notepad để sửa, file khi mở ra sẽ có dạng như hình bên dưới



* Bỏ dòng LastEDS=CANOPEN\_EDS\_MDXPLUS\_Servo\_V1.0.2.eds, sửa FileVersion=1.0 thành FileVersion=1 , sửa FileName=test.dcf thành FileName=test.eds
* Do đây là file ta chủ động mapping nên tại từng giá trị trong file nó sẽ xuất hiện thêm dòng ParameterValue chính là giá trị mà chúng ta đã mapping. Tuy nhiên TIA Portal chỉ hiểu DefaultValue (dành cho file mô tả thiết bị EDS), nó sẽ không hiểu được Parametervalue mà ta đã mapping trước đó. Vì vậy ta cần phải đổi ParameterValue thành DefaultValue để Tia portal hiểu được giá trị mình mapping. Dùng lệnh Ctrl-H để replace toàn bộ kí tự ParameterValue thành DefaultValue . Kết quả nó sẽ như hình bên dưới

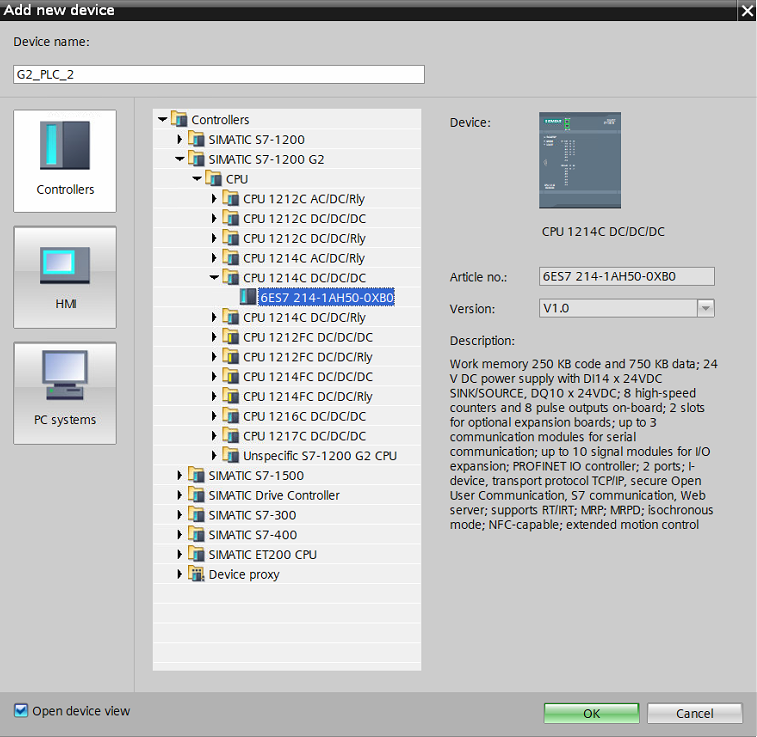


* Sau khi sửa thành công, ta lưu file đó lại để dùng cho các bước tiếp theo

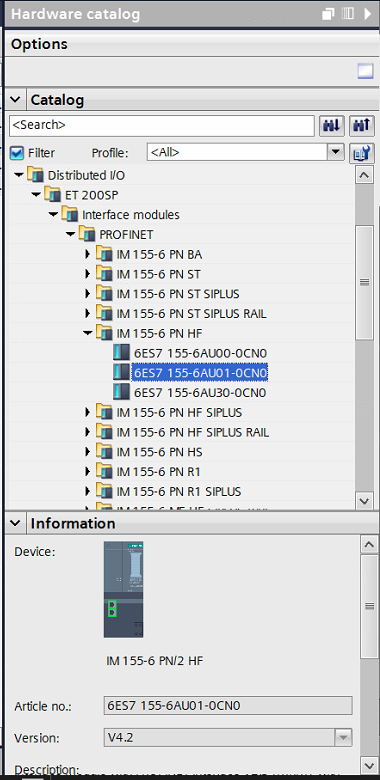
🤏 Thiết lập trên Tia Portal

**BẮT BUỘC PHẢI TẢI PHIÊN BẢN TIA PORTAL V20 Upd4**

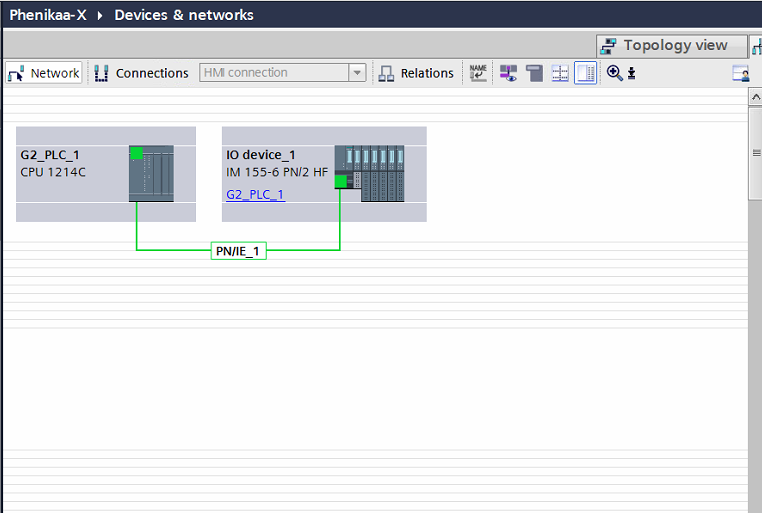
1. **Add new device**



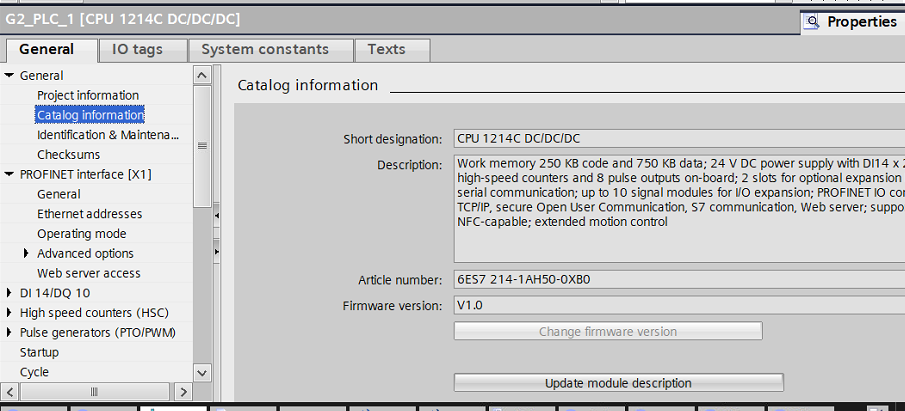
1. **Add module I/O**



1. **Connect giữa 2 module**

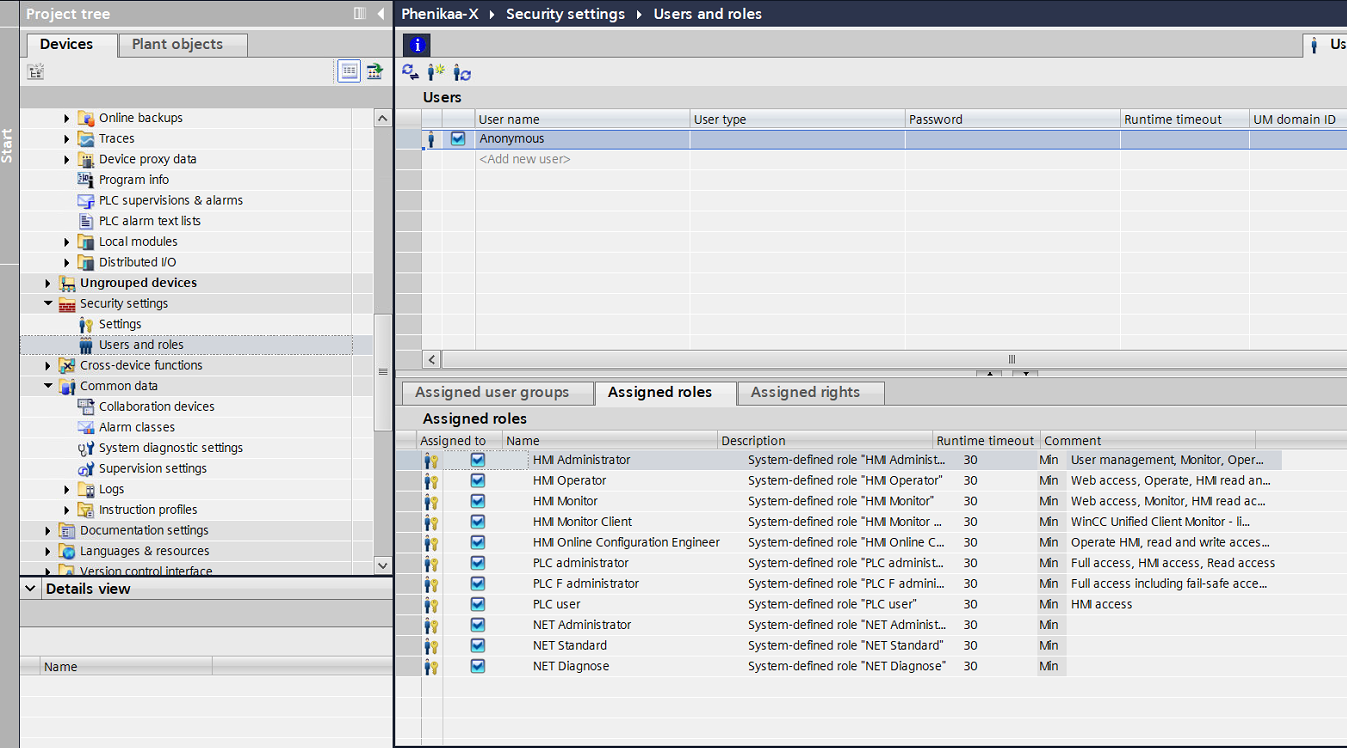


1. **Các thông sô cài cho CPU 1214C**



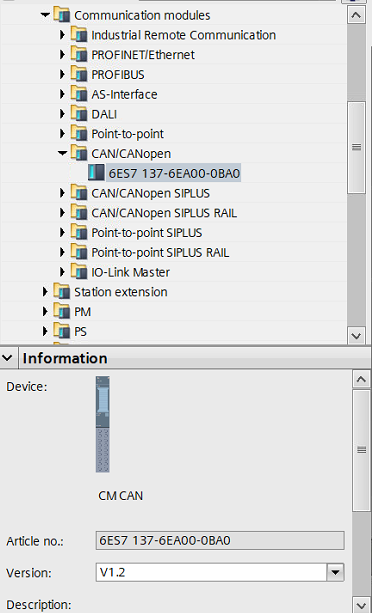


* Cấp quyền

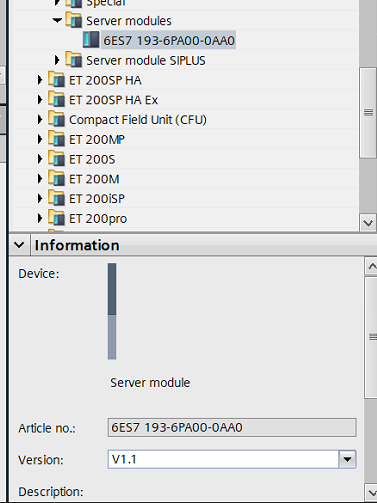


1. **Các thông số cài cho module I/O ET200SP**

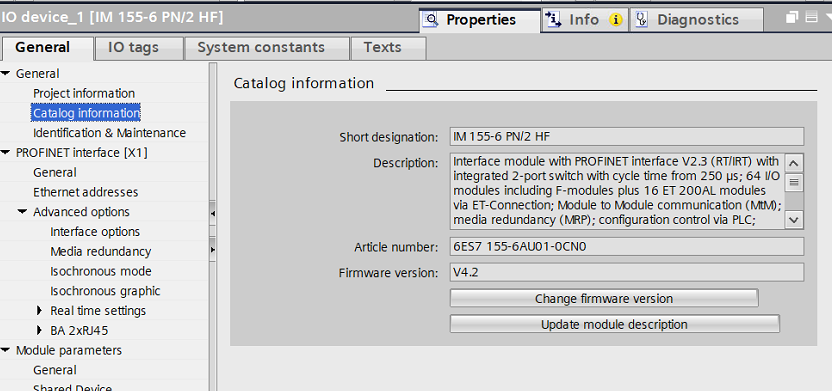
* Add module CM CAN

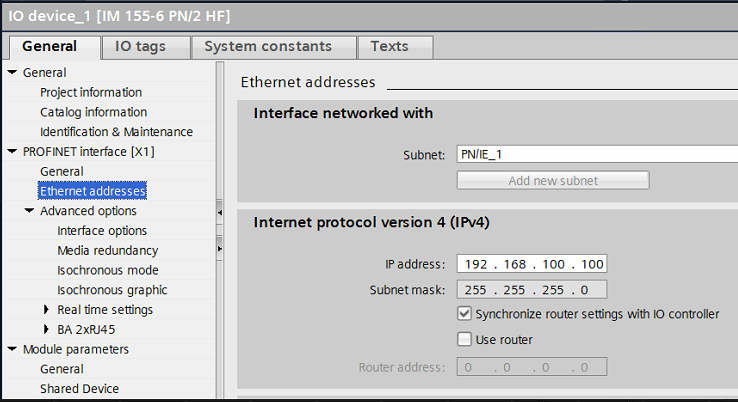


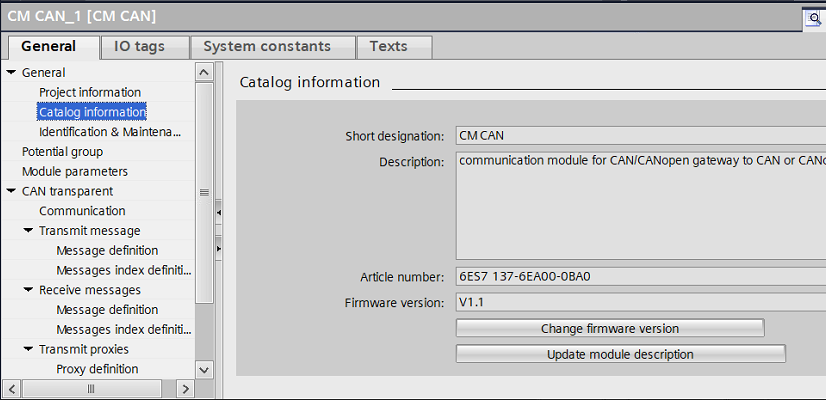
* Add Server module

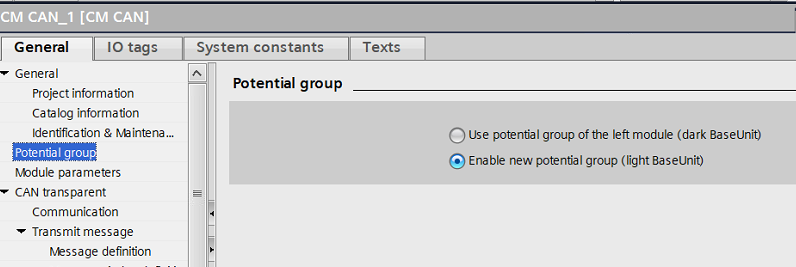


* Cài thông số

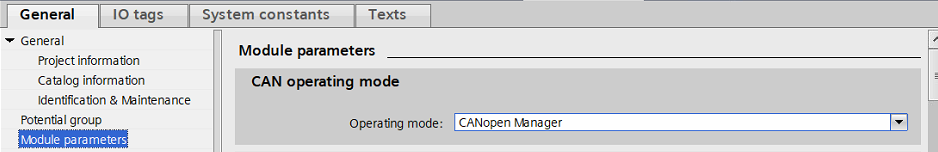




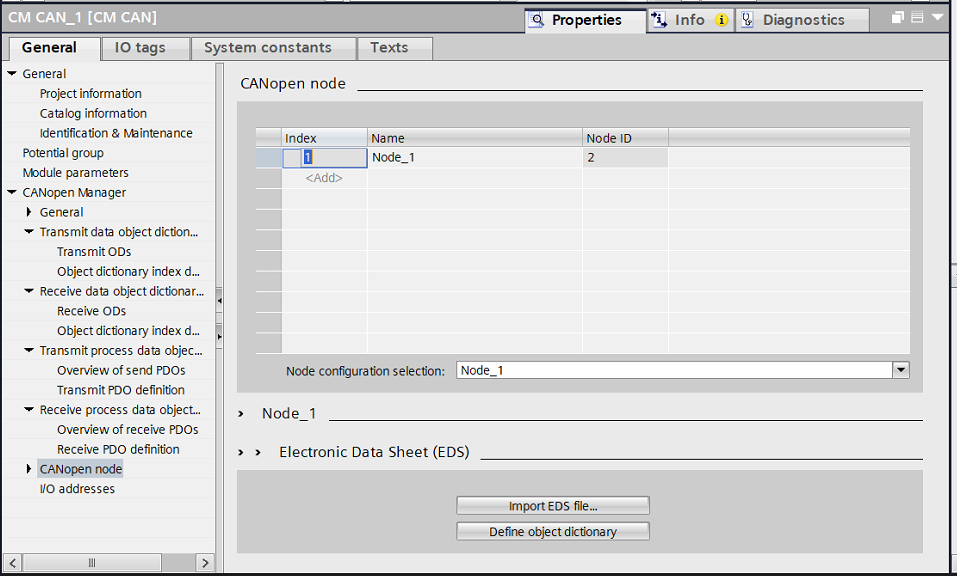




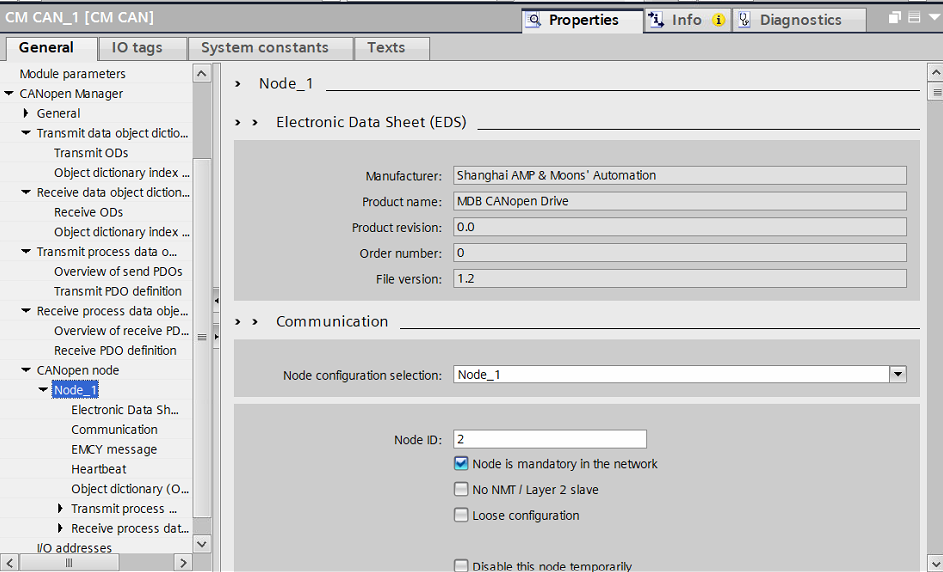
* Chọn chế độ CANopen Manager tại mục Module Parameter



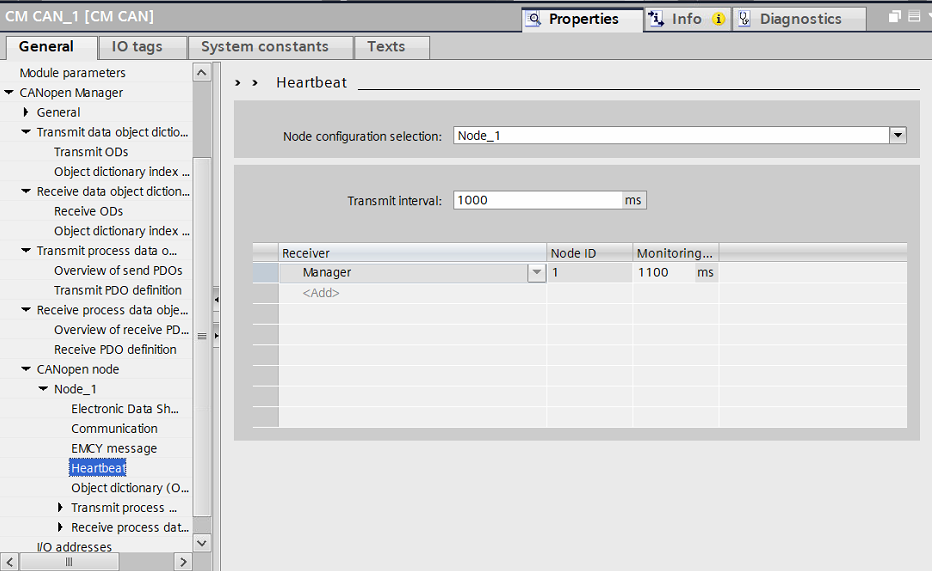
* Tại mục CANopen Node, nhấn add để thêm node

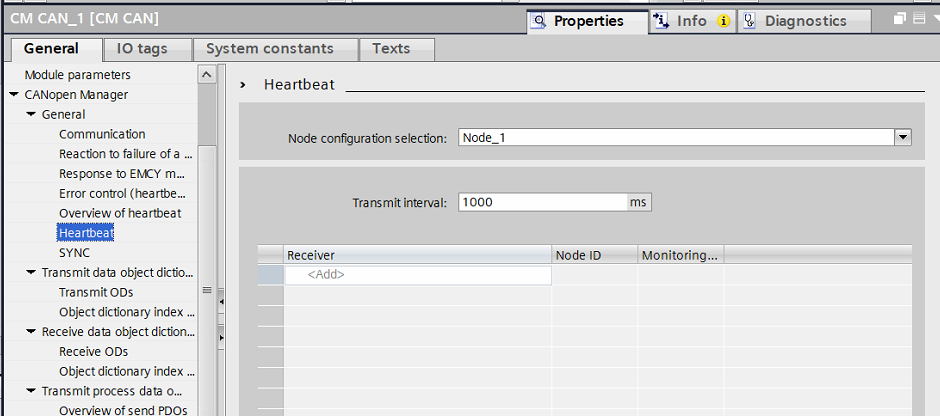


* Sau đó nhấn Import EDS file để import file EDS mình vừa sửa vào, sau khi import kết quả sẽ như hình dưới

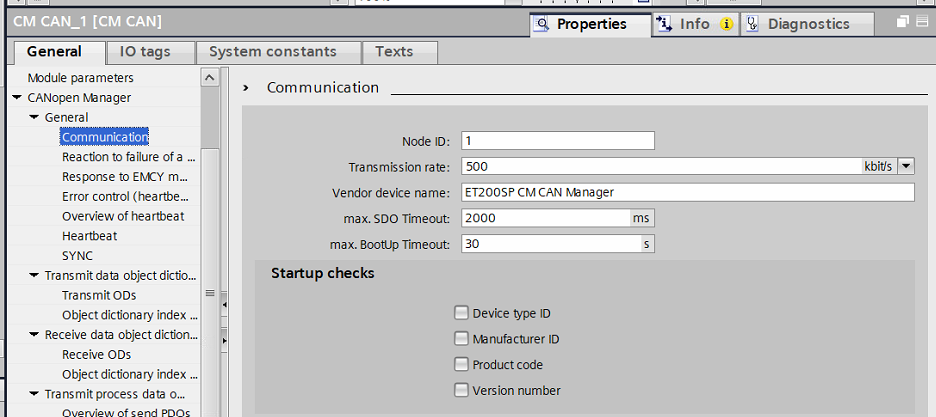


* Tại mục Heartbeat của CANopen node và CANopen Manager, sửa giá trị heartbeat thành 1000 ms theo như đã mapping

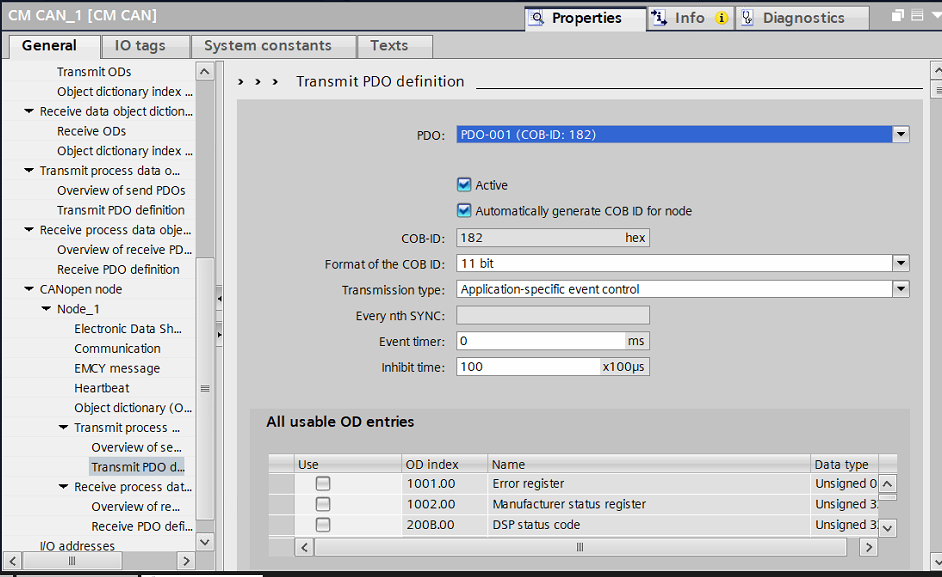


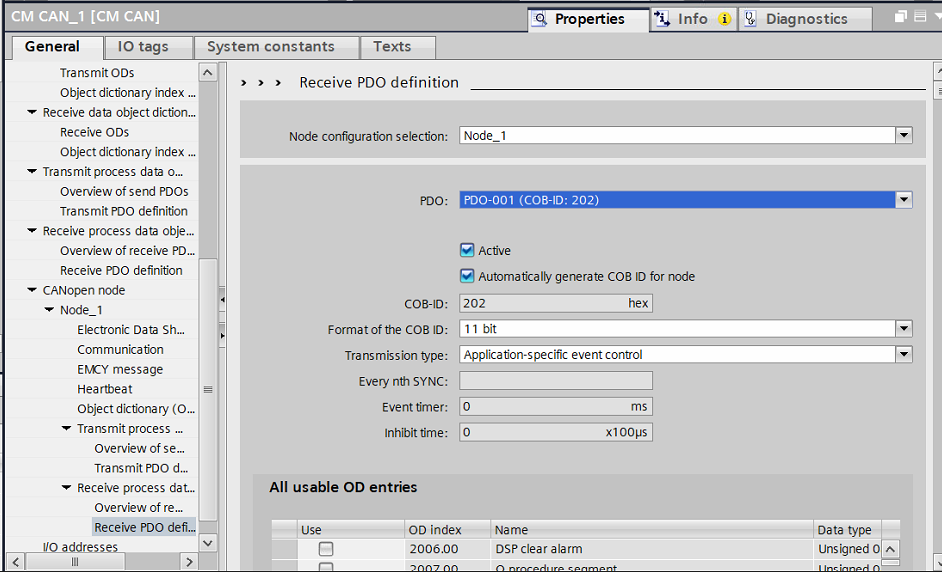


* Đồng thời tại Communication của CANopen Manager, cài giá trị Baudrate đúng với giá trị đã mapping là 500kb/s

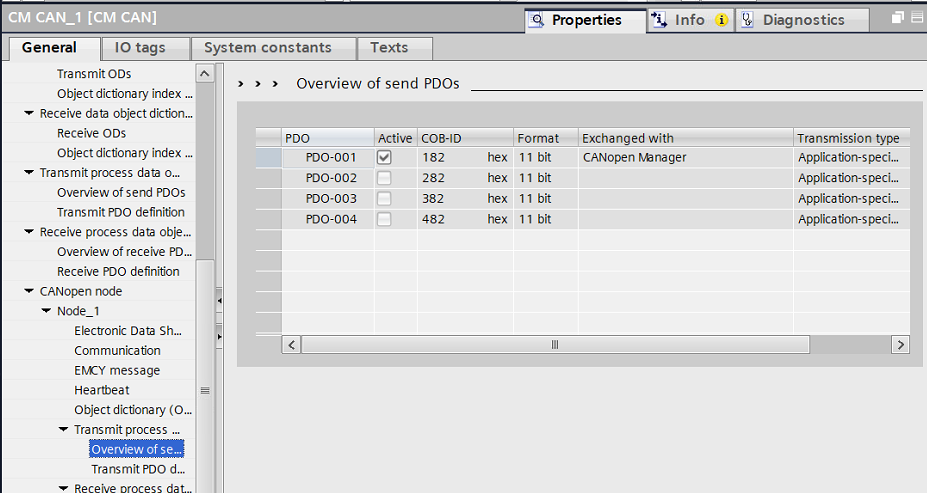


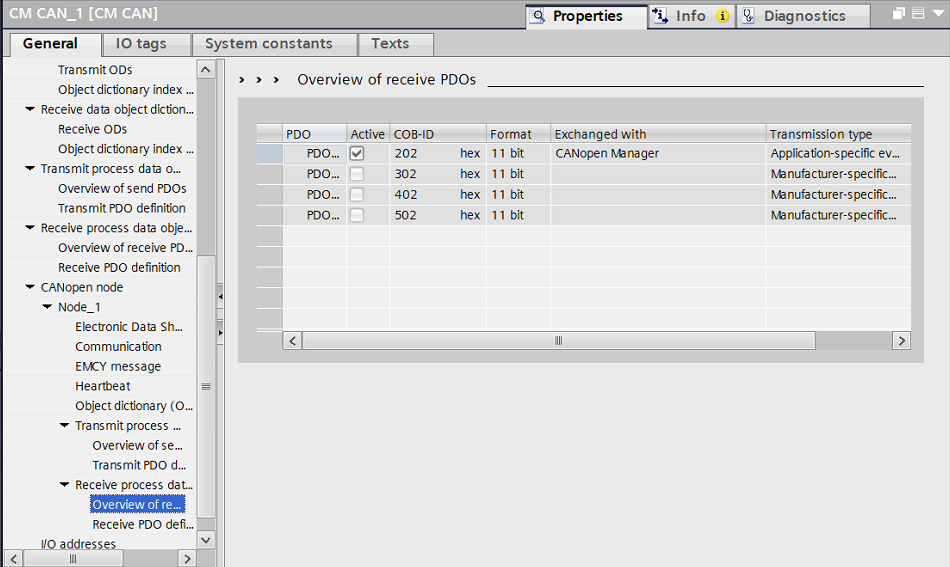
* Lần lượt active TPDO và RPDO ở CANopen Node



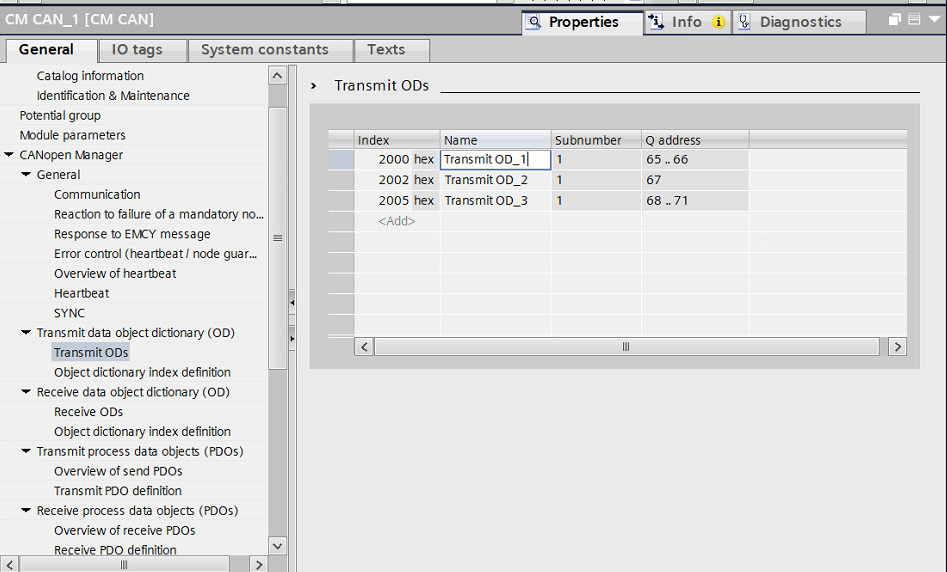


* Kết quả sau khi đã Active

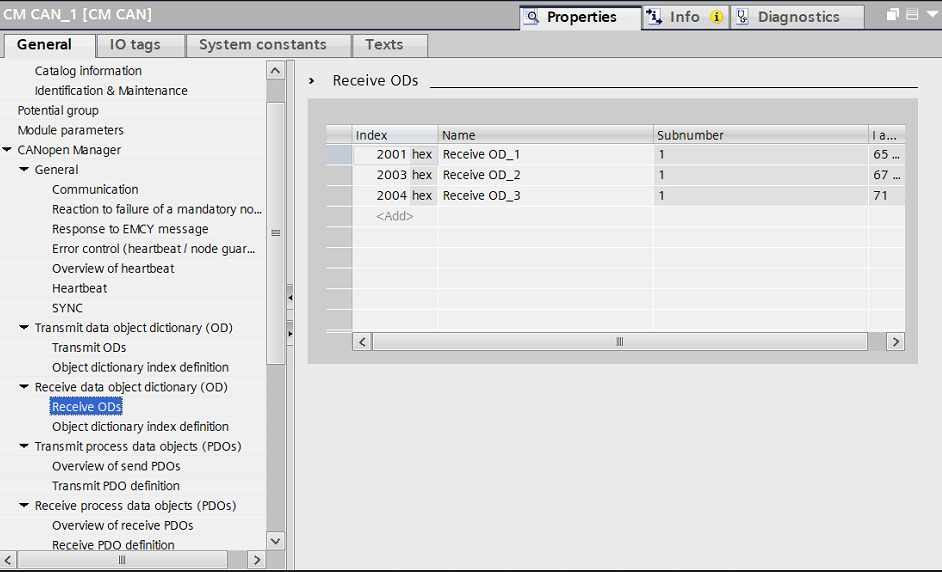




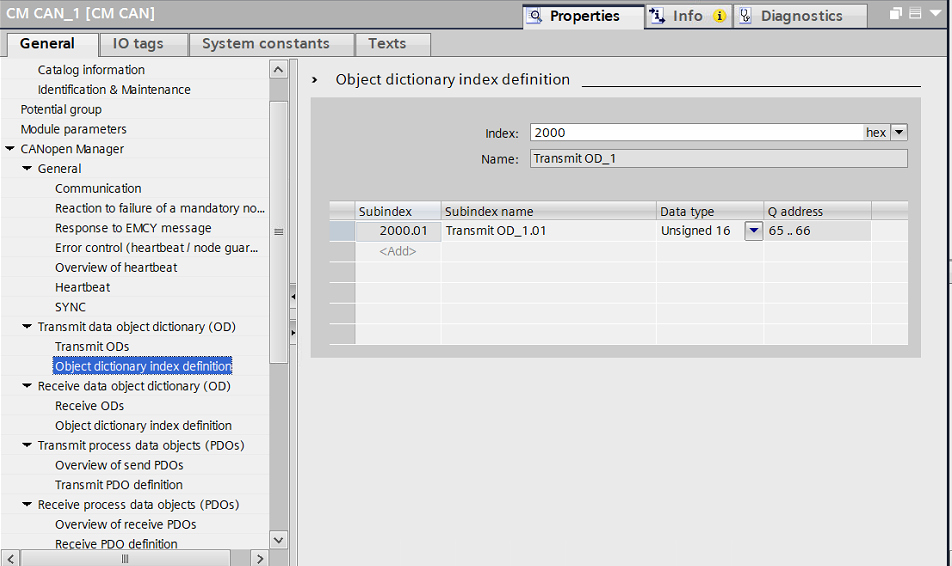
* Tại CANopen Manager -> Transmit data object dictionary -> Transmit ODs . Thêm 3 giá trị như hình dưới đây để phù hợp với 3 giá trị đã được mapping trên PDO

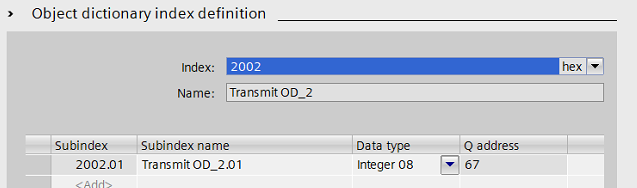


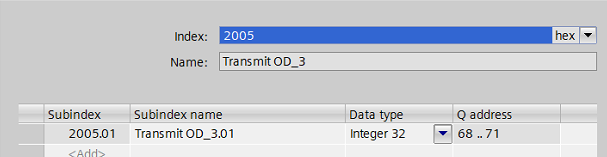
* Tương tự với Receive data object dictionary (OD)

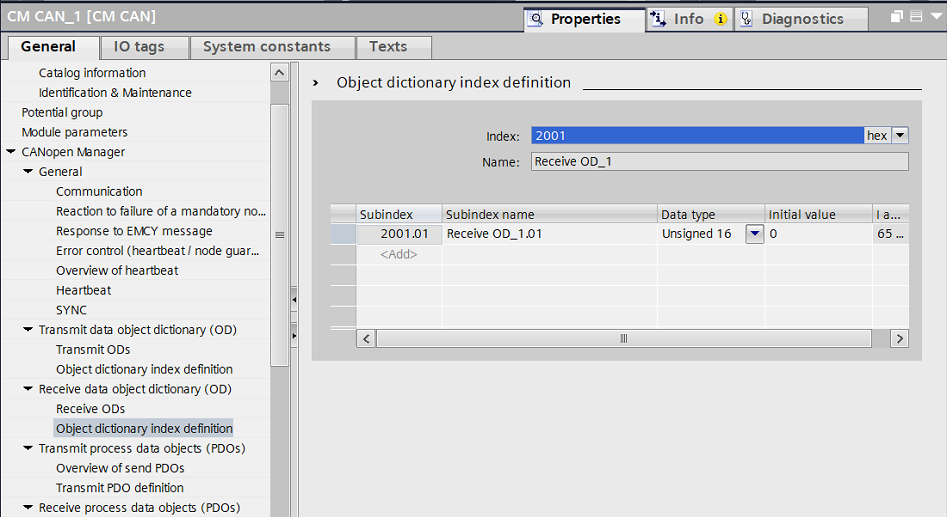


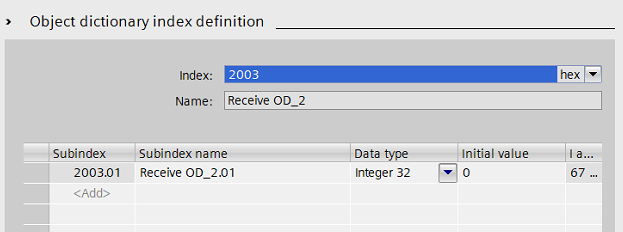
* Lần lượt chỉnh lại Datatype cho từng giá trị cho phù hợp với datatype đã mapping EDS và ở CANopen Node

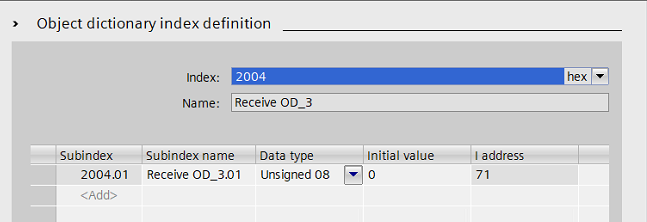




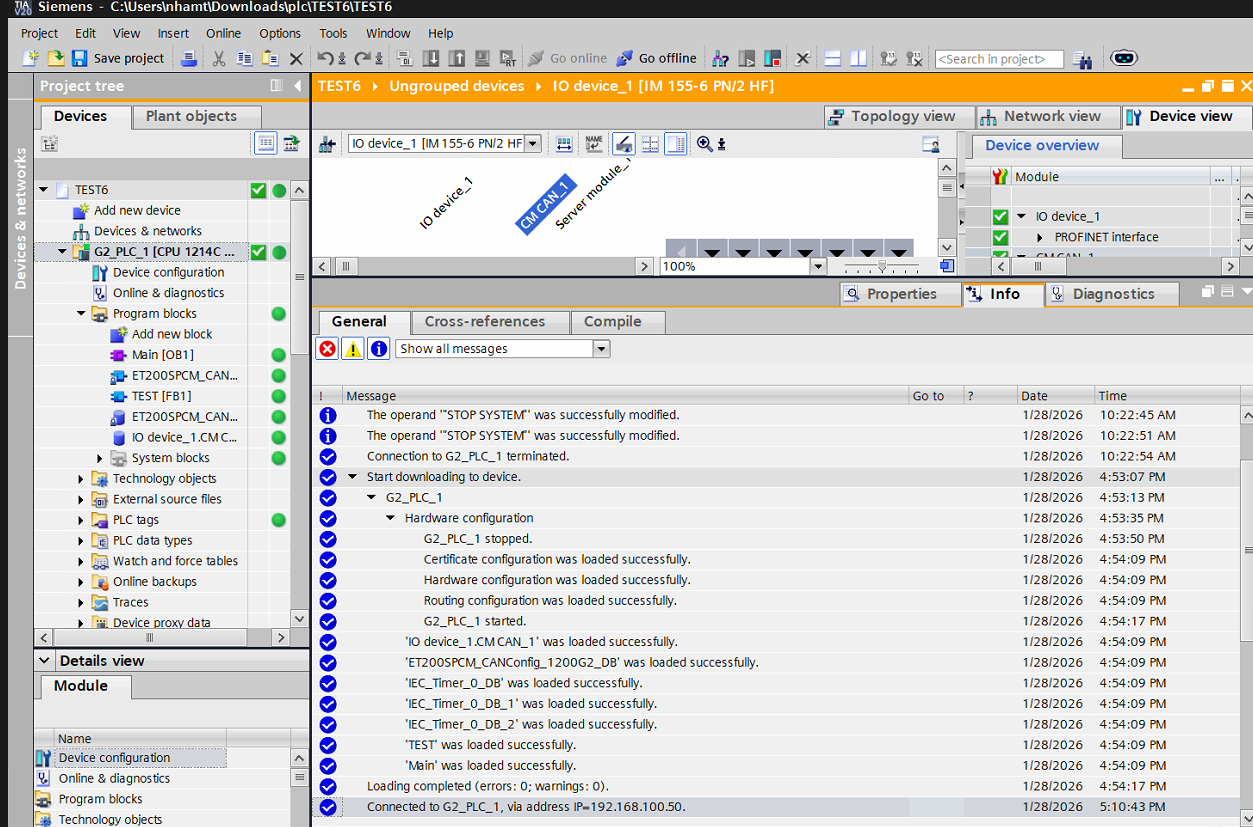






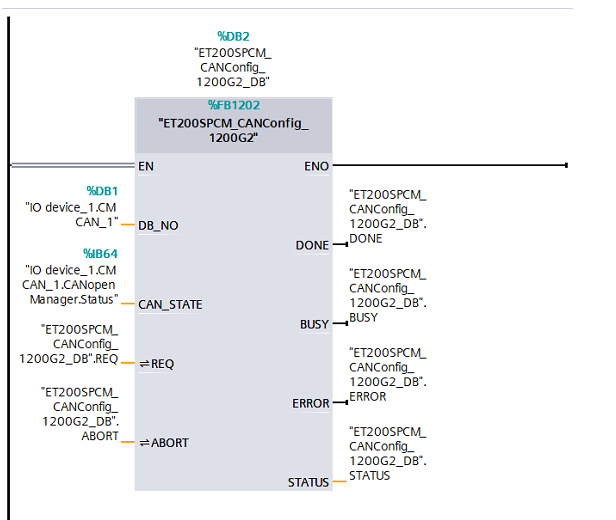


-> Kết quả sau khi thiết lập các tham số thành công



🚖 Lập trình trên Tia Portal

* Cấu hình



1. Các tham số Đầu vào (Input)

| Tham số | Kiểu dữ liệu | Giải thích chi tiết |
| --- | --- | --- |
| **DB\_NO** | Variant / DB\_ANY | **Data Block cấu hình.**<br>Đây là nơi bạn trỏ vào cái Data Block được TIA Portal tự động sinh ra khi bạn cấu hình phần cứng.<br>*(Ví dụ: "IO device\_1.CM CAN\_1")*,. |
| **CAN\_STATE** | Byte | **Byte trạng thái module.**<br>Bạn phải gán **Byte đầu tiên** của vùng địa chỉ Input (%IB...) của module CM CAN vào đây.<br>Hàm dùng byte này để biết module đang ở trạng thái nào (ví dụ: "Not Configured") để quyết định khi nào cần nạp cấu hình,. |

2. Các tham số InOut (Điều khiển)

| Tham số | Kiểu dữ liệu | Giải thích chi tiết |
| --- | --- | --- |
| **REQ** | Bool | **Yêu cầu nạp thủ công (Request).**<br>- Nếu bạn muốn tự tay kích hoạt việc nạp cấu hình, hãy tạo sườn lên (0 → 1) ở chân này.<br>- **Lưu ý:** Nếu bạn đã nối chân CAN\_STATE, khối hàm thường sẽ tự động nạp khi thấy module chưa được cấu hình, bạn không cần kích chân này,. |
| **ABORT** | Bool | **Hủy lệnh.**<br>Kích lên TRUE để dừng quá trình truyền dữ liệu cấu hình ngay lập tức,. |

3. Các tham số Đầu ra (Output - Trạng thái)

| Tham số | Kiểu dữ liệu | Giải thích chi tiết |
| --- | --- | --- |
| **DONE** | Bool | Lên TRUE trong 1 vòng quét khi cấu hình đã được nạp xuống module thành công 100%. |
| **BUSY** | Bool | TRUE khi quá trình nạp đang diễn ra. Đừng làm gì khác với module khi đèn này đang sáng. |
| **ERROR** | Bool | Lên TRUE nếu quá trình nạp bị lỗi. |
| **STATUS** | DWord | **Mã lỗi.**<br>Nếu ERROR = TRUE, chân này sẽ hiện mã Hex để bạn tra cứu nguyên nhân (ví dụ: DB sai, module không phản hồi),. |

4. Cách đấu nối thực tế trong chương trình (OB1)

Để module tự động chạy ngay khi cấp điện mà không cần code phức tạp, bạn hãy đấu nối như sau:

1. **DB\_NO**: Kéo cái System DB của module CM CAN vào (Tìm trong thư mục *System blocks* hoặc gõ tên module như trong Hardware Config).

2. **CAN\_STATE**: Gán địa chỉ %IBx (với x là địa chỉ bắt đầu của module, ví dụ %IB65).

3. **REQ** và **ABORT**: Có thể để trống (hoặc gán biến Tag nếu muốn điều khiển tay).

**Code điều khiển bật tắt động cơ MOONS’**

