

GIẢI PHÁP VIẾT CHƯƠNG TRÌNH ĐIỀU KHIỂN GIÁM SÁT VAN BALLAST TRÊN TÀU THỦY SỬ DỤNG MÔ-ĐUN PLC WINCPU VỚI SỰ HỖ TRỢ CỦA TRÍ TUỆ NHÂN TẠO (AI) CONTROL SOLUTION FOR WRITING PROGRAM OF CONTROLLING AND MONITORING BALLAST WATER ON SHIPS USING PLC WINCPU MODULE WITH SUPPORTED BY ARTIFICIAL INTELLIGENCE (AI)

TRẦN TIẾN LƯƠNG*, TRẦN SINH BIÊN, VŨ THỊ THU

Khoa Điện - Điện tử, Trường Đại học Hàng hải Việt Nam

*Email liên hệ: luongtt.ddt@vimaru.edu.vn

Tóm tắt

Bài báo trình bày một giải pháp viết chương trình điều khiển và giám sát hệ thống van nước ballast trên tàu thủy dựa trên việc tích hợp mô-đun WinCPU thuộc dòng MELSEC-Q Series của Mitsubishi. Thay vì sử dụng PLC truyền thống với các ngôn ngữ lập trình LAD hoặc SCL, hệ thống mới được xây dựng bằng ngôn ngữ Visual Basic trong môi trường Visual Studio, tận dụng thư viện lập trình QBFFunc32 và sự hỗ trợ của trí tuệ nhân tạo (AI). Giải pháp cho phép điều khiển và giám sát, xử lý sự kiện thời gian thực.

Từ khóa: WinCPU, thư viện QBFFunc32, điều khiển nước ballast, AI, PLC Melsec-Q.

Abstract

This paper presents a solution for writing programs of controlling and monitoring the ballast valve system on ships based on the integration of the WinCPU module from the Mitsubishi MELSEC-Q Series. Instead of using traditional PLCs with programming languages such as LAD or SCL, the proposed system is developed in Visual Basic within the Visual Studio environment, leveraging the QBFFunc32 programming library and the support of artificial intelligence (AI). The solution enables real-time control, monitoring, and event handling.

Keywords: WinCPU, QBFFunc32 library, Ballast water control, AI, PLC Melsec-Q.

1. Đặt vấn đề

Trong ngành Hàng hải, hệ thống nước ballast (ballast water system) đóng vai trò sống còn trong việc đảm bảo tính ổn định, an toàn và hiệu quả vận hành của tàu thủy. Khi tàu không chở hoặc chở ít hàng, trọng tâm tàu có xu hướng cao, làm mất cân bằng và dễ gặp rủi ro trong điều kiện thời tiết xấu hoặc khi

chuyển hướng. Để khắc phục điều này, các kết chứa nước ballast được sử dụng nhằm điều chỉnh trọng lượng và cân bằng tàu, giúp duy trì độ chìm, tư thế và khả năng điều động ổn định.

Ngoài việc đảm bảo ổn định thân tàu, hệ thống ballast còn được sử dụng trong các thao tác như:

- Hạ tàu xuống mực nước phù hợp khi ra/vào cảng;
- Điều chỉnh món nước đáp ứng yêu cầu hàng hải;
- Hỗ trợ quá trình xếp/dỡ hàng hóa hiệu quả hơn.

Trong các tàu đang hoạt động hiện nay, các hệ thống đang sử dụng PLC để điều khiển và kết hợp với một HMI hoặc một máy tính trung tâm để giám sát. Hệ thống hoạt động với hai phần là CPU của PLC (Programmable Logic Controller) để thu thập dữ liệu qua các mô-đun vào/ra và phần SCADA (Supervisory Control and Data Acquisition) riêng. Với việc thay thế CPU thường thành một mô-đun WinCPU, hệ thống vẫn hoàn toàn kết hợp được với thiết bị phần cứng cũ và tận dụng được khả năng của WinCPU để mở ra các tính năng mới như:

Tận dụng được khả năng tính toán, lưu trữ dữ liệu trên nền Window.

Chạy được các bài toán nâng cao: Tối ưu lịch bơm ballast, phân tích bất thường, học máy (machine learning).

WinCPU thường đi kèm với giao diện HMI/SCADA chạy trực tiếp trên máy tính, cho phép dễ dàng điều khiển, truy cập qua ứng dụng hoặc Web, phù hợp với khả năng giám sát từ xa cho tàu khi cần. Đặc biệt có thể bảo trì phần mềm từ xa.

Việc lập trình WinCPU sử dụng các ngôn ngữ như C#, C++, VB.NET trên phần mềm Visual Studio thay cho việc lập trình PLC thông thường với SCL, FBD, LAD,... Đặc biệt với các ngôn ngữ bậc cao, ta có thể dễ dàng tạo ra được các hàm chức năng với sự hỗ trợ của AI.

Dễ kết nối với camera, cảm biến, hệ thống giám sát qua OPC UA (Open Platform Communications Unified Architecture), MQTT (Message Queuing

Telemetry Transport), Modbus TCP,...

Chuyển từ PLC truyền thống sang WinCPU không chỉ giúp hệ thống ballast gọn hơn, mạnh hơn, thông minh hơn, mà còn mở ra khả năng tích hợp AI, giám sát từ xa, đưa con tàu tiến gần hơn đến mô hình “smart ship” hiện đại.

2. Giới thiệu về mô-đun WinCPU

WinCPU là một mô-đun điều khiển thông minh thuộc dòng MELSEC-Q Series của Mitsubishi, được thiết kế để hoạt động như một bộ xử lý nhúng (embedded CPU) tích hợp hệ điều hành Windows Embedded Standard 2009. Khác với PLC truyền thống chỉ chuyên xử lý logic điều khiển, WinCPU còn cho phép xử lý dữ liệu, tính toán phức tạp, kết nối mạng và chạy phần mềm như một máy tính công nghiệp thực thụ [1].

Mô-đun này hỗ trợ cấu hình multi-CPU, cho phép chia sẻ công việc giữa CPU điều khiển logic (PLC) và CPU xử lý dữ liệu (WinCPU), tạo nên hệ thống điều khiển và tính toán liền mạch. Nó có thể khởi động từ ổ SSD tích hợp hoặc thẻ CF, đồng thời trang bị hai cổng LAN tốc độ cao, năm cổng USB, và nhiều giao diện mở rộng khác giúp dễ dàng kết nối với thiết bị ngoại vi, hệ thống giám sát hoặc nền tảng IoT [1].

Nhờ sử dụng bộ xử lý Intel® Atom™ N450 tiết kiệm điện năng, cùng khả năng chạy phần mềm chuẩn Windows, WinCPU mở ra khả năng ứng dụng các thuật toán AI, giao diện điều khiển nâng cao, và kết nối từ xa cho các hệ thống tự động hóa công nghiệp hiện đại [1], [4].



Hình 1. Mô-đun WinCPU

Để sử dụng mô-đun MELSEC WinCPU trong hệ thống điều khiển, ta cần tiến hành một trình tự cấu hình thiết bị theo quy trình bao gồm ba giai đoạn chính:

1. Cấu hình ban đầu hệ thống WinCPU: Thiết lập tiện ích, thông số vận hành và giao tiếp với các mô-đun mở rộng. Việc thiết lập cấu hình hệ thống sẽ được

thực hiện trên tiện ích *MELSEC WinCPU Setting Utility, CC-Link Utility*. Hệ thống cũng thực hiện kiểm tra liệu có cần thay đổi các thông số cấu hình hay không, bao gồm cài đặt hệ thống, địa chỉ I/O, cấu hình CPU và các tham số truyền thông.

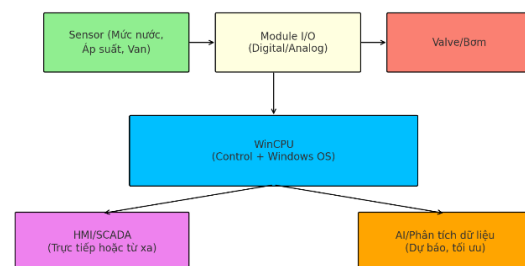
2. Khởi tạo chương trình điều khiển: Bao gồm việc mở bus, khởi động bộ đếm cơ chế giám sát Watchdog Timer và lập trình chức năng điều khiển thiết bị ngoại vi. Chương trình được người dùng tạo ra trên phần mềm với các hàm chức năng từ thư viện QBFFunc32.dll, QBFFunc32.c, QBFFunc32.vb trong môi trường phát triển như Visual Studio. Việc sử dụng WDT giúp theo dõi các lỗi hệ thống như treo chương trình, quá tải xử lý hoặc mất kết nối thiết bị, từ đó kịp thời đưa ra phản ứng thích hợp nhằm đảm bảo an toàn cho hệ thống điều khiển.

3. Giám sát và xử lý lỗi thời gian thực: thông qua việc đọc trạng thái mô-đun, kiểm tra lỗi, đặt lại hoặc dừng WDT và đóng bus an toàn.

3. Xây dựng hệ thống điều khiển ballast cho tàu HPC Sunrise

Trong bài báo này, nhóm tác giả dựa trên hệ thống ballast của tàu HPC Sunrise để thiết kế lại một hệ thống thay thế sử dụng WinCPU. Trong cấu trúc tàu chở hàng rời như tàu HPC Sunrise, các kết chứa nước ballast được điều khiển bởi một hệ thống thủy lực tích hợp - cho phép đóng/mở các van liên quan thông qua điều khiển từ xa bằng van điện từ và bộ chấp hành thủy lực [3]. Hệ thống bao gồm:

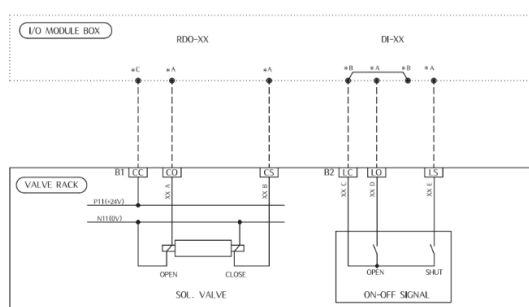
- Bộ nguồn thủy lực (Hydraulic Power Pack) tạo áp lực từ 110 - 130 bar để cấp cho các van.
- Tủ điều khiển van thủy lực, vận hành bởi van điện từ 24V DC có báo vị trí.
- 48 van được điều khiển từ xa, qua bộ chia áp suất và đường ống thủy lực.
- Cơ cấu bơm tay và bơm khẩn cấp được tích hợp để đảm bảo vận hành trong mọi tình huống, kể cả khi mất điện.



Hình 2. Sơ đồ kết nối điều khiển nước ballast sử dụng WinCPU

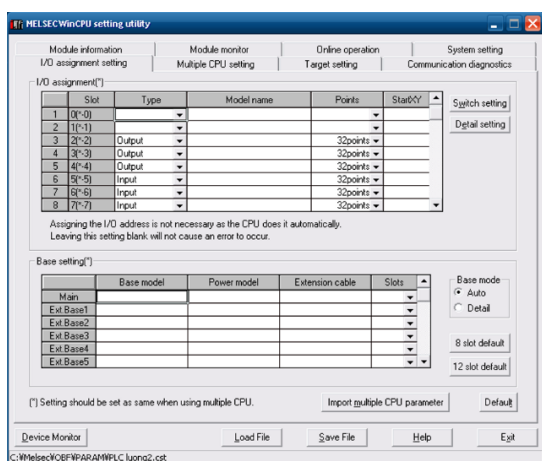
Sơ đồ khối cho việc kết nối điều khiển nước ballast sử dụng WinCPU thể hiện trên Hình 2. Các tín hiệu về cảm biến mức nước, áp suất và vị trí van được kết nối tới các mô-đun vào DI của PLC - Q. Trong khi đó, tác tín hiệu điều khiển sẽ được đưa ra thông qua các mô-đun tín hiệu ra số DO.

Hệ thống cho phép điều khiển van qua các cuộn đóng và mở van. Đồng thời giám sát trạng thái van nhờ các tín hiệu phản hồi. Sơ đồ tín hiệu điều khiển và giám sát của van tới hệ thống được mô tả như Hình 3. Với mỗi van, ta sẽ có 2 tín hiệu vào số để giám sát trạng thái đóng hết và mở hết của van. Cùng với đó, mỗi van sẽ có 2 đầu ra điều khiển cho việc đóng và mở van [3].



Hình 3. Sơ đồ kết nối tín hiệu điều khiển và giám sát van

Sau khi lắp đặt các mô-đun để tạo ra số lượng đầu vào/ra đáp ứng được yêu cầu của hệ thống. Ta tiến hành xác định và khai báo cấu hình trạm cho WinCPU trên phần mềm MELSEC WinCPU Setting Utility. Với ba mô-đun DI32 và DO32 để điều khiển và giám sát 48 van ballast cài đặt địa chỉ như Hình 4.

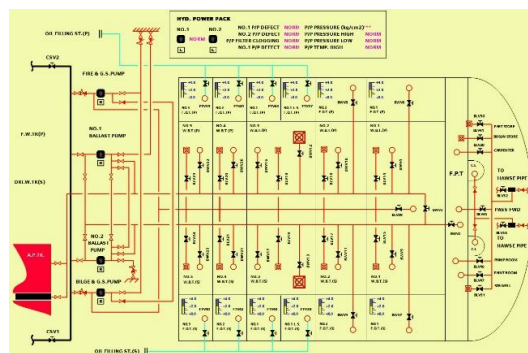


Hình 4. Khai báo cấu hình trạm cho mô-đun

WinCPU

Trên phần mềm Visual Basic, tác giả tiến hành xây dựng một giao diện giám sát. Trong đó 48 van ballast từ van 0 đến van 47 sẽ được sử dụng Picture và xây dựng thành một mảng với 48 phần tử có tên Val(x) - x

từ 0 đến 47. Hệ thống điều khiển sẽ tập trung vào hai phần chính: Phần hiển thị trạng thái của van và phần điều khiển van. Giao diện điều khiển và giám sát được thể hiện như Hình 5.



Hình 5. Giao diện điều khiển và giám sát

Để triển khai xây dựng chương trình, nhóm tác giả đã phân chia hệ thống thành các các chương trình con. Sau khi xác định các yêu cầu chức năng, AI được sử dụng để thiết kế các chương trình con tương ứng. Các chương trình con này sau đó được tích hợp và gọi trong chương trình chính để tạo thành một hệ thống điều khiển hoàn chỉnh. Thuật toán tổng thể của chương trình điều khiển và giám sát hệ thống nước ballast trên tàu được minh họa tại Hình 6.



Hình 6. Thuật toán chương trình điều khiển và giám sát

Dựa trên cấu trúc thuật toán đã được thiết kế, quá trình lập trình được thực hiện bằng cách phân chia hệ thống thành các chương trình con chuyên biệt. Mỗi

chương trình con đảm nhận một chức năng cụ thể, được xây dựng thông qua các truy vấn lập trình tương tác với ChatGPT (phiên bản o3-mini-high), tập trung vào điều khiển logic và tự động hóa [2], [5]. Do giới hạn phạm vi trình bày trong khuôn khổ bài báo, nhóm tác giả chỉ minh họa một số chương trình con quan trọng thay vì toàn bộ hệ thống. Cụ thể, các chương trình con được xây dựng bao gồm:

+ Câu lệnh cho ChatGPT-o3-Mini viết chương trình khởi tạo kết nối Bus giữa các mô-đun trong hệ thống MELSEC-Q: Sử dụng thư viện QBFFun32.vb để viết chương trình mở kết nối bus trong quá trình khởi động hệ thống. Khi kết nối thành công, hệ thống tự động kích hoạt bộ định thời Timer_500ms. Trạng thái kết nối được hiển thị thông qua nhãn LblStatus.. Kết quả mã nguồn trả về từ AI là:

```
Private Sub Form_Load()
    Ret = QBF_Open(unit, path)
    If Ret = 0 Then
        LblStatus.Caption = "Status: OK"
        Timer1_500ms.Enabled = True
    Else
        LblStatus.Caption = "Status: fail"
    End If
End Sub
```

+ Câu lệnh cho ChatGPT-o3-Mini viết chương trình đọc tín hiệu trạng thái van từ mô-đun đầu vào số (DI) của PLC: “Sử dụng hàm QBF_X_in_bit trong thư viện QBFFunc32.vb để truy xuất tín hiệu đầu vào. Số lượng van sử dụng biến ValveCount để có thể tùy chỉnh. Trạng thái van được lưu trữ vào mảng một chiều InputValState. Trong trường hợp có lỗi, mã lỗi được hiển thị tại nhãn LblErr. Kết quả mã nguồn trả về từ AI là:

```
Private Sub ReadInputFromX()
    Dim i As Integer, data As Byte
    For i = 0 To ValveCount * 2 - 1
        Ret = QBF_X_In_Bit(path, i + StartX, data)
        If Ret <> 0 Then
            LblErr.Caption = "Error Read X: " & (i + StartX) & " ret:" & Ret
            InputValState(i) = -1
            Exit For
        Else
            InputValState(i) = data
        End If
    Next i
```

End Sub

+ Câu lệnh cho ChatGPT-o3-Mini viết chương trình ngắt tín hiệu điều khiển khi van đã đạt trạng thái đóng/mở hoàn toàn: Viết chương trình cập nhật điều tín hiệu điều khiển của đầu ra. Khi van đã đến giới hạn đóng thì ngắt đầu ra đóng. Khi van tới giới hạn mở thì ngắt đầu ra mở. Trạng thái đầu ra của van lưu trong mảng OutputValveState.

```
Private Sub UpdateValveLogic()
    Dim i As Integer
    For i = 0 To ValveCount - 1
        idxInClose = 2 * i
        idxInOpen = 2 * i + 1
        If InputValState(idxInClose) = 1 Then
            OutputValveState(idxInClose) = 0
        End If
        If InputValState(idxInOpen) = 1 Then
            OutputValveState(idxInOpen) = 0
        End If
    Next i
End Sub
```

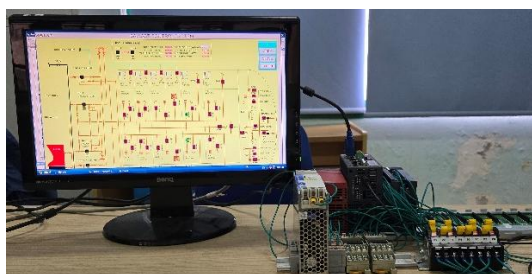
+ Câu lệnh cho ChatGPT-o3-Mini viết chương trình điều khiển van từ giao diện giám sát điều khiển (HMI): Hệ thống gồm 48 van, mỗi van được đại diện bởi một phần tử trong mảng Val kiểu hình ảnh Picture. Khi người dùng tương tác với một van bất kỳ, hệ thống hiển thị ba nút lệnh: CmdOpen, CmdClose, CmdCancel, cho phép thay đổi giá trị điều khiển tương ứng trong biến OutputValveStatus.” Kết quả mã nguồn trả về từ AI là:

```
Private Sub UpdateValveDisplay()
    Dim i As Integer, idxInClose As Integer, idxInOpen As Integer
    Dim inClose As Integer, inOpen As Integer
    Dim outClose As Byte, outOpen As Byte
    blinkState = Not blinkState
    For i = 0 To ValveCount - 1
        idxInClose = 2 * i
        idxInOpen = 2 * i + 1
        inClose = InputValState(idxInClose)
        inOpen = InputValState(idxInOpen)
        outClose = OutputValveState(idxInClose)
        outOpen = OutputValveState(idxInOpen)
        If outOpen = 1 Then
            Val(i).Picture = ImageList1.ListImages(2).Picture
        ElseIf outClose = 1 Then
```

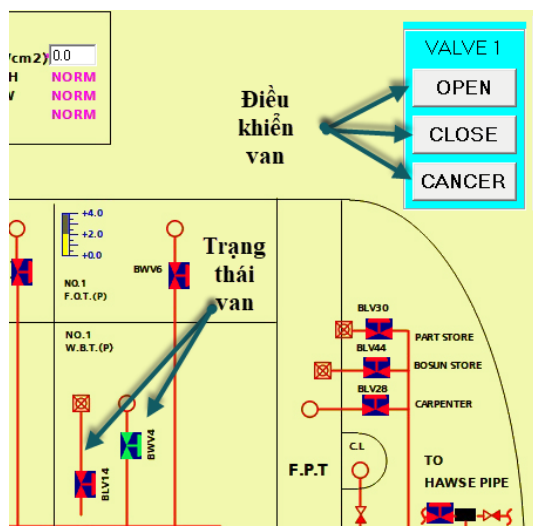
```

Val(i).Picture = ImageList1.ListImages(1).Picture
ElseIf inOpen = 1 Then
    Val(i).Picture = ImageList1.ListImages(2).Picture
ElseIf inClose = 1 Then
    Val(i).Picture = ImageList1.ListImages(1).Picture
End If
If blinkState And (outOpen=1 Or outClose=1) Then
    Val(i).Picture = ImageList1.ListImages(3).Picture
End If
Next i
End Sub

```



Hình 7. Kết quả chạy thử nghiệm phần mềm



Hình 8. Kết quả hiển thị điều khiển giám sát Van

Một chương trình điều khiển và giám sát cho 48 van đã được xây dựng với các mã nguồn có được ở trên. Giao diện giám sát và hệ thống điều khiển hoạt

động của van hoạt động tốt. Trạng thái van hiển thị trên giao diện giám sát như hình 8. Van thể hiện màu xanh khi mở và màu đỏ khi đóng. Khi đang trong quá trình đóng/mở van sẽ nhấp nháy với màu tương ứng. Mỗi khi kích vào một van bất kỳ, bảng điều khiển tương ứng sẽ hiện ra tại góc trên bên phải của màn hình. Tại đây cho phép người vận hành thực hiện các thao tác đóng/mở van.

4. Kết luận

Bài báo đã trình bày một giải pháp sử dụng trí tuệ nhân tạo AI viết chương trình điều khiển và giám sát hệ thống van nước ballast trên tàu thủy sử dụng mô-đun WinCPU kết hợp với lập trình Visual Basic và thư viện chức năng QBFFunc32 bằng ngôn ngữ VB. Giải pháp sử dụng AI hỗ trợ thực thi chương trình cho các PLC giúp giảm thời gian viết chương trình điều khiển của các kỹ sư tự động, thay vì viết chương trình và gỡ lỗi, người kỹ sư sẽ chuyển sang định hướng cấu trúc và chức năng của hệ thống để AI viết chương trình tương ứng.

Thông qua việc tận dụng sức mạnh tính toán của WinCPU cùng sự hỗ trợ từ mô hình ngôn ngữ ChatGPT, nhóm tác giả đã xây dựng thành công một chương trình điều khiển 48 van điện từ, có tính tùy biến linh hoạt, giao diện trực quan và dễ dàng mở rộng. Các hàm chức năng như mở kết nối bus, đọc tín hiệu đầu vào, cập nhật điều khiển đầu ra, và xử lý tương tác người dùng đều được tự động hóa với sự hỗ trợ của AI, giúp rút ngắn đáng kể thời gian phát triển phần mềm.

Lời cảm ơn

Nghiên cứu này được tài trợ bởi Trường Đại học Hàng hải Việt Nam trong đề tài mã số: DT24-25.61.

TÀI LIỆU THAM KHẢO

- [1] MELSECWinCPU module User's Manual, nguồn <http://www.mitsubishielectric.com>.
- [2] Sử dụng ChatGPT, nguồn <https://help.openai.com>.
- [3] Valve Remote Control system DZ06H-15/16/11, nguồn www.seilseres.com
- [4] iQ-R MelsecWinCPU module Programming manual, nguồn <http://www.mitsubishielectric.com>.
- [5] Nguyễn Thị Kim Trang, Hoàng Đức Hải, Nguyễn Thị Ngọc Mai, Nguyễn Hữu Anh (2005), *Microsoft Visual Basic 6.0 Và Lập trình cơ sở dữ liệu*, NXB Lao Động - Xã hội.

Ngày nhận bài:	05/04/2025
Ngày nhận bản sửa:	16/04/2025
Ngày duyệt đăng:	18/04/2025