đề tài của em là THIẾT KẾ, THI CÔNG BỘ ĐIỀU KHIỂN PCS

Trong thời đại 4.0, công nghiệp hóa và số hóa là xu hướng không thể đảo ngược.

PLC HMI là thiết bị điều khiển và hiển thị thông tin quan trọng trong các hệ thống tự động hóa.

Tuy nhiên, PLC HMI có những hạn chế như giá thành cao, khả năng mở rộng thấp và giao diện người dùng không thân thiện.

Để khắc phục những hạn chế đó, đề tài này dùng vi điều khiển và làm hình cảm ứng thay thế PLC HMI trên hệ PCS là một giải pháp tiềm năng.

Có những ưu điểm như giá thành rẻ, khả năng tùy biến cao.

Đề tài này có tính cấp thiết cao vì nó góp phần nâng cao hiệu quả và tiết kiệm chi phí cho nhà trường và doanh nghiệp trong thời đại 4.0.

Để đáp ứng được tính cấp thiết đó: mục tiêu đề tài của em là thay thế PLC và màng hình HMI của PLC bằng vi điều khiển với màng hình cảm ứng

Đầu tiên, em sẽ nói khái quát về hệ PCS của trường ta, hệ PCS là một hệ thống để thí nghiệm điều khiển closed loop control (cụ thể là thuật toán PID) và 2-step control

Các đối tượng thí nghiệm của hệ là:

+ Mức nước (Level)

+ Lưu lượng (flow rate)

+ Áp suất (pressure)

+ Nhiệt độ (temperature)

Tuy nhiên giới hạn đề tài của em thì chỉ Thí nghiệm điều khiển: closed loop control (PID), và 3 Đối tượng thí nghiêm:

+ Mức nước (Level)

+ Lưu lượng (flow rate)

+ Áp suất (pressure)

So sánh giữa PLC HMI và Vi điều khiển HMI

- PLC HMI có tính ổn định cao, dễ lắp đặt và sử dụng, nhưng có chi phí cao, khả năng mở rộng hạn chế và khó thay đổi giao diện sau khi lập trình, to lớn cồng kềnh.

- Vi điều khiển HMI có chi phí thấp, khả năng tùy biến cao, hỗ trợ nhiều tính năng đồ họa và hiệu ứng, nhỏ gọn, nhưng cần kỹ năng lập trình cao.

Tiếp theo là phần chon phương án MCU HIM:

Dựa trên các tiêu chí:

- Khảo sát thị trường.

- Nền tảng hỗ trợ thiết kế và lập trình UI.

- Hiệu năng

- Cấu hình

Sau khi khỏa sát thị trường Việt Nam thì có thể mua được 3 loại MCU HMI sau:

Arduino, rẻ, nhiều thư viện, nhưng ko có nền tảng hỗ trợ thiết kế và lập trình UI. Hiệu năng kém, cấu hình thấp.

Màng hình Nextion, có nền tảng hỗ trợ thiết kế và lập trình UI , giá thành cao, yêu cầu thêm MCU để thực thi backend, hiệu năng kém.

Stm32F469-DISC Board, có nền tảng hỗ trợ thiết kế và lập trình UI, Hiệu năng cao, cấu hình cao, phù hợp với ứng dụng HIM.

Tiếp theo là phân tích hệ PCS:

Dựa trên:

- Sơ đồ khối của điều khiển vòng kính của hệ.

- Các tính hiệu điện

Sơ đồ hệ kính của hệ gồn có:

cài setpoint từ touch screen -> setPoint qua bộ tổng, trừ với feedback.

sau đó được đưa vào bộ điều khiển PID. output của PID sẽ truyền đến Processs

phản hồi của procees sẽ được sensor thu thập và truyền về bộ tổng.

và tiếp tục vòng lập như thế.

Sau đó là phân tích các dạng tính hiệu điện trên hệ.

Sau khi khảo sát thì đi đến kết luận, trên hệ có 4 dạng tính hiệu:

AI, AO, DI, DO

Sau đó là chọn module chuyển đổi tính hiệu.

Vì các driver và cảm biến PCS được dùng các dụng cụ công nghiệp, vì thế để MCU có thể điều khiển cũng như đọc được tình hiệu trên hệ PCS thì cần phải qua các module chuyển đổi tính hiệu.

Đối với DI, chọn Module::Opto, DO chọn Module::Relay, AO Module::pwmToAnalog, AI chọn Mạch chia áp tự thiết kế.

Về nguồn điện của của MCU và các module thì em sẽ Tận dụng bộ nguồn 24V có sẵn trên hệ PCS để lấy nguồn 24v, kết hợp với Module::DC 24V to DC 5V để có nguồn 5V để cấp cho MCU và các module cần nguồn 5V khác.

Sau đây là sơ đồ khối.

Tiếp theo là phần đồ họa.

Stm32 có hỗ trợ nền tảng hỗ trợ thiết kế và lập trình UI được gọi là STM32TouchGfx.

Đây là nền tảng tương đối mới mẻ và chưa phổ biến ở Việt Nam và nước ngoài.

Tiếp theo là phần định hướng thiết kế của phần mềm:

"bare-metal" là gì?

Phân tích đặt điểm của hệ thống:

Chương trình phức tạp.

Hệ thống có nhiều trạng thái.

Hệ thống có nhiều event.

Yêu cầu thời gian thực.

Yêu cầu xử lý đa nhiệm và chia sẻ tài nguyên.

đi đến kết luận là: việc lập trình bare metal sẽ rất khó khăn cho phát triển và bảo trì.

Viết firmware cho các peripheral:

Nói sơ về firmware: Firmware là phần mềm chạy trên phần cứng, điều khiển các chức năng và hoạt động của các thiết bị ngoại vi.

Cấu hình các ngoại vi theo đúng yêu cầu, theo đúng các pin.

Gồm có các firmware:

External Interrupt: để nhận digital In

PWM

ADC

Digital Output

sau đó là ta đi phát trển các task của phần mền:

Việc áp dụng hệ điều hành dẫn đến việc cần thiết kết các task cho hệ thống.

Nói sơ qua về task: Task là một đơn vị cơ bản của hệ điều hành thời gian thực (RTOS). Một task có thể được hiểu như một chương trình con độc lập, thực hiện một nhiệm vụ cụ thể.

Vậy ta có các task như là: task tính toán PID, task đọ và gửi dữ liệu analog, task xử lý đồ họa.

các task này sẽ hoạt động song song với nhau.

phần cơ khí: thiếu.

Cơ khí: không thay đổi kết cấu của hệ, để dàng chuyển đổi giữa điều khiển PLC và MCU

Tới đây là ta đã hoàn thành hệ thống:

Tiếp đến là khâu đánh giá chất lượng hệ thống,

Xem xét tính đặt thù của hệ thống thì, thệ thống có thế sai sót ngay cả khi đúng đắng về mặt logic:

nhưng hệ thống có thể có sai số, và ko tin cậy gây ra bởi vì hiệu năng hệ thống, linh kiện, độ ưu tiên của các task.

Ta đưa ra được các tiêu chí đánh giá:

- Đánh giá độ tuyến tính của Module tự thiết kế
- Chất lượng biểu đồ

Độ tuyến tính của module mạch chia ap tự thiết kế:

Ap dụng lý cơ sở lý thuyết về mạch chia ap, dung sai của điện trở, và đinh nghĩa hàm số tuyến tính

thì có để đi đến kết luận là mạch chia áp này tuyến tính.

Về chất lượng biểu đồ thì có các lỗi sai khả dĩ như sau?

Thứ 1: Sai lệch giữa thời gian thực và biểu đồ. do biểu đồ cập nhật bị trễ quá nhiều so với thời gian thật

Thư 2: Update dữ liệu biểu đồ sai lệch về thời gian với thực thế. Đúng về giá trị, nhưng sai về mặt thời gian.

Thứ 3: Mất đi đặt tính quan trọng của biểu đồ, do tần số lấy mẫu quá thấp.

Vì thế cần phân tích đường đi của dữ liệu từ cảm biến đến biểu đồ.

Từ cảm biết qua senso đến khi MCU đọc được giá trị cảm biến là mất 11.11 nanoSec

Còn lại là từ khi pid được tính toán đến khi vẽ dc biểu đồ lên màng hình

Để biết được khoảng thời gian này, ta sẽ cài nền tảng Segger system view vào phần mềm của MCU.

Segger system view là: Segger system view là một công cụ phân tích và kiểm tra hiệu năng của các hệ thống nhúng. Nó cho phép người dùng theo dõi các sự kiện trong hệ thống, như các tác vụ, ngắt, thời gian chạy, và các thông số khác.

Ngoài ra còn so sánh với biểu đồ vẽ trên máy hiện sóng.

Biểu đồ trên máy hiện sóng và biểu đồ mà ta vẽ được là gần như tưng tự nhau.

Khoảng thời gian từ lúc chạy đền lúc dừng cũng chính xác.

Điều đó đi đến kết luận là hệ thống thiết kế, đảm bảo được tính trung thực của biểu đồ.

Và cuối cùng là phần thực nghiệm:

Em đã thực nghiệm điều khiển cả 3 hệ: level, flow, và pressure.

Nhưng vì đảm bảo thời lượng của phần thuyết trình, em sẽ chỉ trình bàn trường hợp điều khiển pid hệ lưu lượng.

Vào màng hình điều khiển pid.

Đóng/mở các van sao cho nước được bơm chạy qua cảm biến lưu lượng.

Trên màng hình, mở digital 1 để chọn bơm ở chế độ analog.

setPoint là 200 ml/p

kP = 1, cảm biến dead zone.

kP = 2, cảm biến đã thu được lưu lượng rõ nét hơn.

kP = 4 và kp = 8, đối với sai số xác lập giảm, hệ giao động nhỏ. chưa đạt được xét point

kP = 1; Ti = 0.1. Triệt tiêu được sai số xác lập. hệ giao động lớn, không ổn định

kP = 1; Ti = 1; Triệt tiêu được sai số xác lập. hệ giao động lớn, nhưng tắt dần

kP = 1; Ti = 2;, Ti = 3, thì đồ vọt lố càng giảm, tắt dần càng nhanh, feedback càng bám setpoint.

Segger system view là một công cụ phân tích và kiểm tra hiệu năng của các hệ thống nhúng. Nó cho phép người dùng theo dõi các sự kiện trong hệ thống, như các tác vụ, ngắt,

thời gian chạy, và các thông số khác. Ngoài ra, nó cũng cung cấp các tính năng như đồ thị, bảng, biểu đồ, và bộ lọc để giúp người dùng hiểu rõ hơn về hoạt động của hệ thống.

Segger system view được dùng bởi các nhà phát triển và kỹ sư phần mềm nhúng, đặc biệt là những người làm việc với các hệ điều hành thời gian thực (RTOS) hoặc các ứng dụng đa luồng. Bằng cách sử dụng segger system view, họ có thể tối ưu hóa hiệu năng, khắc phục lỗi, và kiểm tra chất lượng của các sản phẩm nhúng.