**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**Trường Cơ Khí**

\*\*\*\*\*

****

Bài tập lớn môn : Kỹ thuật lập trình trong Cơ diện tử

Giảng viên : TS. Trương Công Tuấn

Đề tài : Cánh tay Robot gắp vật tự động

điều khiển bằng Joystick

Sinh viên thực hiện :

Trần Tiến Anh – 20184346

1. Giới thiệu chung.
2. Tóm tắt đề tài:

* Cánh tay Robot 3 bậc tự do,có thể được điều khiển bằng tay qua Joystick.
* Ngoài ra cánh tay Robot còn có thể tự động gắp và thả vật tại 2 vị trí.
* 2 vị trí này hoàn toàn có thể được set up bằng Joystick hoặc qua giao diện Winform .
* Cánh tay Robot cũng liên tục gửi tín hiệu tọa độ và hiển thị lên giao diện điều khiển theo thời gian thực.

1. Các linh kiện sử dụng:

* 1 Arduino Uno R3
* 2 Modul Joystick
* 4 Động cơ Servo

1. Cách thức điều khiển Robot:

* Thực hiện chuyển đổi giữa chế độ Auto, Control qua 2 nút nhấn Auto và Control trên giao diện WindowForm.
* Sử dụng 2 Joystick để điều khiển Robot trong chế độ Control.
* Trong chế độ Auto, ta phải set up trước cho Robot 2 vị trí điểm đầu và điểm cuối, cánh tay Robot sẽ thực hiện liên tục gắp, thả vật qua 2 vị trí này.
* Để set up 2 vị trí cho cánh tay Robot, ta có thể dùng 2 cách, đó là:

+ Cách 1: Dùng Joystick điều khiển cánh tay Robot qua các vị trí cần set up và nhấn các nút SetPosition1ByJoyStick, SetPosition2 ByJoyStick trên giao diện để lưu 2 vị trí đầu và cuối.

+ Cách 2: Trực tiếp điền tọa độ cần set up lên các textBox tương ứng và nhấn các nút SetPosition1, SetPosition2 trên giao diện điều khiển.

1. Đặt vấn đề.

* Cánh tay robot là thiết bị có tầm quan trọng và được sử dụng rộng rãi trong lĩnh vực tự động hóa, cánh tay robot trong hầu hết các trường hợp được lập trình và sử dụng để thực hiện các nhiệm vụ cụ thể, do đó thường có quỹ đạo chuyển động cố định, không dễ dàng thay đổi. Ngoài ra hầu hết các Robot arm đều có kết cấu phức tạp, khó điều khiển.
* Trước vấn đề đó, em đã có ý tưởng về 1 Robot arm có thể dễ dàng điều khiển cũng như thay đổi quỹ đạo trong khi đang hoạt động.

1. Sơ đồ giải thuật.

Arduino:

Sử dụng timer số 2 để gửi tọa độ Robot về giao diện điều khiển mỗi 0,4 giây 1 lần

Kết thúc vòng Loop

Signal = “fir”

Thực hiện chế độ Auto

Kiểm tra chuỗi signal nhận được từ WindowForm

Thực hiện chế độ Control

Lưu vị trí được gửi từ Windowform làm vị trí cuối

Lưu vị trí được gửi từ Windowform làm vị trí đầu

Lưu vị trí hiện tại làm vị trí cuối

Lưu vị trí hiện tại làm vị trí đầu

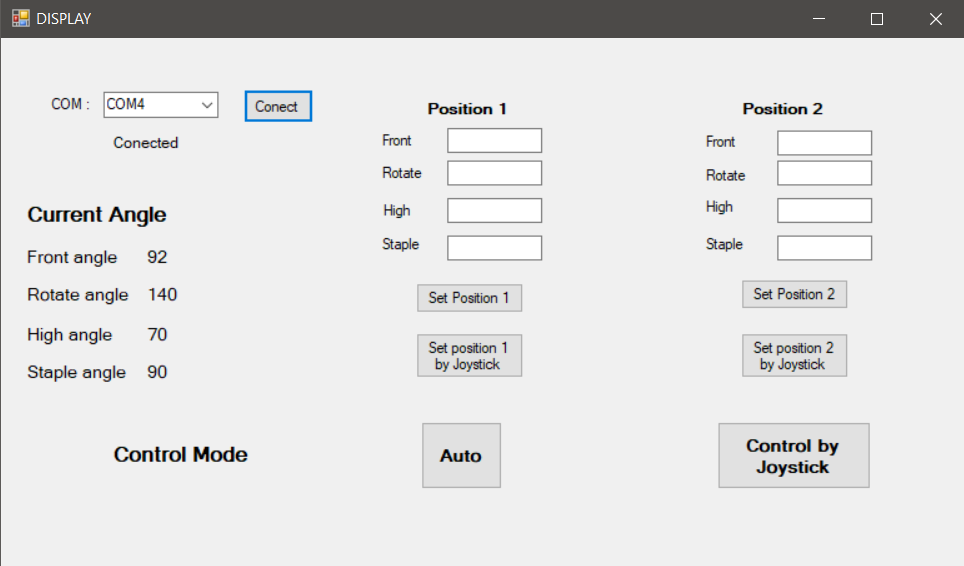
Signal bắt đầu = dấu \*

Signal = “sec”

Signal bắt đầu = dấu #

signal = “aut” signal = “con”

1. Giao diện điều khiển.



Các nút nhấn thay đổi giữa 2 chế độ

Các nút nhân để thiết lập vị trí 2 điểm đầu, cuối trong chế độ Auto

Tọa độ cánh tay robot theo thời gian thực

Vị trí điểm đầu trong chế độ Auto

Vị trí điểm cuối trong chế độ Auto

1. Thuật toán sử dụng.
2. Thuật toán điều khiển cánh tay Robot bằng Joystick:

* Joystick bao gồm 2 biến trở có giá trị từ 0 => 1023, khi Joystick ở trạng thái bình thường thì giá trị của các biến trở vào khoảng 520 => 540.
* Lợi dụng đặc điểm này, em sẽ cài đặt khi biến trở có giá trị lớn hơn 800 thì servo tương ứng sẽ quay 1 góc tăng lên 2 độ và delay 0,03s để cho Robot chuyển động mượt mà hơn. Ngược lại, khi biến trở có giá trị nhỏ hơn 200 thì servo tương ứng sẽ quay 1 góc giảm đi 2 độ và delay 0,03s.

1. Thuật toán chuyển đổi giữa 2 chế độ Auto và Control:

* Sử dụng 1 biến toàn cục là biến Flag.
* Khi Flag = True thì Robot sẽ hoạt động trong chế độ Auto, ngược lại khi Flag = False thì sẽ hoạt động trong chế độ Control.
* Mỗi khi bắt đầu vòng loop trong Arduino, ta sẽ đọc tín hiệu Serial gửi từ giao diện điều khiển, và nếu tín hiệu là “aut” thì Flag = True -> Robot chạy trong chế độ Auto, nếu tín hiệu là “con” thì Flag = False -> chuyển sang chế độ Control.

1. Thuật toán xử lý chuỗi:

* Giao tiếp giữa Arduino và Laptop là giao tiếp Serial (UART) qua dây cáp. Điều này có nghĩa là tại 1 thời điểm thì 1 thiết bị chỉ có thể nhận được 1 tín hiệu hoặc gửi đi 1 tín hiệu, mà thông tin tọa độ của Robot lại gồm 4 tọa độ (các góc xoay trên dưới, trước sau, trái phải và góc kẹp của tay máy). Nếu gửi 4 tín hiệu cùng 1 lúc sẽ khó có thể phân biệt được tín hiệu nào ứng với tọa độ nào, dễ bị loạn tín hiệu.
* Do đó em có 1 ý tưởng, đó là ghép 4 tọa độ cần gửi thành 1 chuỗi và gửi đi, sau đó ở thiết bị nhận tín hiệu sẽ thực hiện tách chuỗi nhận được thành 4 chuỗi tín hiệu và sử dụng.

1. Cài đặt Timer:

* Để gửi tín hiệu tọa độ của tay máy trong thời gian thực về giao diện điều khiển, em phải sử dụng tới Timer (do các thuật toán điều khiển đều nằm trong vòng lặp loop, nếu viết thuật toán gửi tín hiệu cũng trong vòng loop thì sẽ không đảm bảo tín hiệu được gửi đi liên tục).
* Trên Arduino uno R3 có 3 bộ Timer là Timer0, Timer1 và Timer2. Do Timer0 dùng cho hàm delay(), Timer1 dùng cho việc điều khiển động cơ Servo nên em chỉ còn Timer 2 để sử dụng.
* Timer2 là Timer 8 bit, có nghĩa là thanh ghi của nó có thể đếm được từ 0 ->255. Tần số của Arduino là 16 MHz, prescaler lớn nhất là 1024 => chu kì lớn nhất của Timer2 là 1024/16000000\*256 = 0,016384s. Đây là khoảng thời gian quá nhỏ, dễ gây giật lag và điều khiển không ổn định.
* Do đó, để mỗi 0,4s gửi 1 tín hiệu như ý muốn, em sử dụng thêm 1 biến toàn cục là biến count. Biến count sẽ chạy từ 1 đến 50, mỗi khi timer tràn thì biến count tăng thêm 1 và khi count = 50 mới gửi tín hiệu đi và set lại cho count = 1. Bằng cách này, ta có thể giảm tần số gửi tín hiệu đi thêm 50 lần nữa.

1. Kết luận.

* Cánh tay Robot đã hoạt động được như ý muốn.
* Tuy nhiên vẫn còn có hiện tượng giật lag và hoạt động chưa được ổn định, nguyên nhân có thể do phần cứng chưa đảm bảo chất lượng.
* Trong tương lai có thể phát triển sản phầm thêm bằng cách gắn thêm các cảm biến tiệm cận, camera giúp cánh tay Robot có thể thực hiện nhiều tác vụ hơn, thông minh hơn.

1. Tài liệu tham khảo:

* Timer/Counter trên Arduino: <http://arduino.vn/bai-viet/411-timercounter-tren-avrarduino>
* Các phương thức về SerialPort trên .NET FrameWork: <https://learn.microsoft.com/en-us/dotnet/api/system.io.ports.serialport?view=dotnet-plat-ext-7.0>