

TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP.HCM
KHOA CÔNG NGHỆ CƠ KHÍ



BÁO CÁO HỌC PHẦN

Thực hành tự động hóa và robot

GVHD: LƯƠNG QUỐC VIỆT

SVTH: TÔN TRẦN TUẤN KIỆT	2025210343
VÕ MINH TIẾN	2025224413
NGUYỄN NHẬT QUANG	2025211066
NGUYỄN LÊ NHỰT LỄ	2025211783

TP. Hồ Chí Minh, tháng 08 năm 2025

TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP.HCM
KHOA CÔNG NGHỆ CƠ KHÍ



BÁO CÁO HỌC PHẦN
THỰC HÀNH TỰ ĐỘNG HÓA VÀ ROBOT

GVHD: LƯƠNG QUỐC VIỆT

SVTH: TÔN TRẦN TUẤN KIẾT 2025210343

VÕ MINH TIẾN 2025224413

NGUYỄN NHẬT QUANG 2025211066

NGUYỄN LÊ NHỰT LỄ 2025211783

TP. Hồ Chí Minh, tháng 08 năm 2025

Mục lục

CHƯƠNG 1. TỔNG QUAN.....	1-1
1.1. Giới thiệu sơ lược về Robot:	1-1
1.1.1. Cấu tạo:	1-1
1.1.2. Chức năng:	1-1
1.1.3. Điều khiển:	1-2
1.1.4. Ứng dụng:.....	1-2
1.2. Mục tiêu cần đạt của bài thuyết trình:	1-2
CHƯƠNG 2. NGUYÊN LÝ HOẠT ĐỘNG.....	2-1
2.1. Kết cấu cơ khí.....	2-1
2.2. Điện – Điện tử	2-2
2.2.1. Động cơ Servo.....	2-2
2.2.2. Điều khiển Robot	2-3
2.3. Truyền thông	2-8
CHƯƠNG 3. THIẾT KẾ GIAO DIỆN ĐIỀU KHIỂN	3-1
3.1. Thiết kế Giao diện điều khiển	3-1
3.1.1. Phần mềm Qt Designer	3-1
3.1.2. Thiết kế giao diện.....	3-1
3.2. Lập trình thiết kế giao diện trên VS Code.....	3-3
CHƯƠNG 4. SƠ ĐỒ GIẢI THUẬT CỦA ĐIỀU KHIỂN ROBOT	4-1
4.1. Sơ đồ giải thuật:	4-1
4.2. Lưu đồ giải thuật:	4-1
CHƯƠNG 5. KẾT QUẢ.....	5-1
CHƯƠNG 6. KẾT LUẬN.....	6-1
Mục lục.....	6-1
6.1. Lập trình thuật toán và hoàn thiện.....	Error! Bookmark not defined.

CHƯƠNG 1. TỔNG QUAN



1.1. Giới thiệu sơ lược về Robot:

1.1.1. Cấu tạo:

Cánh tay robot là một thiết bị cơ khí được thiết kế để thực hiện các nhiệm vụ giống như cánh tay của con người, thường được ứng dụng trong các lĩnh vực như sản xuất, y tế, nghiên cứu, và dịch vụ. Dưới đây là một số thông tin cơ bản về cánh tay robot:

- **Khung và động cơ:** Cánh tay robot thường bao gồm các khớp, bộ điều khiển, động cơ và khung. Các khớp cho phép di chuyển và xoay, trong khi đó động cơ cung cấp lực cần thiết để thực hiện các chuyển động.
- **Cảm biến:** Nhiều cánh tay robot được trang bị cảm biến để nhận biết môi trường, cảm nhận lực và hồi tiếp, giúp cải thiện độ chính xác và khả năng tương tác.
- **Gắp hoặc tay:** Phần cuối của cánh tay robot thường được trang bị một bộ gắp hoặc tay để thực hiện các tác vụ như nhấc, xoay và di chuyển các vật thể.

1.1.2. Chức năng:

- **Lắp ráp:** Cánh tay robot thường được sử dụng trong dây chuyền lắp ráp để thực hiện các nhiệm vụ như gắn các linh kiện, gói hàng, và kiểm tra chất lượng.
- **Phẫu thuật:** Trong y tế, cánh tay robot được sử dụng trong phẫu thuật ít xâm lấn, giúp bác sĩ thực hiện các thao tác chính xác hơn.
- **Khám phá:** Một số cánh tay robot được sử dụng trong các nhiệm vụ khám phá, chẳng hạn như trên các tàu vũ trụ hoặc trong môi trường nguy hiểm, như nhà máy hạt nhân hoặc khu vực thiên tai.

1.1.3. Điều khiển:

- Cánh tay robot có thể được điều khiển bằng cách sử dụng nhiều phương pháp khác nhau, bao gồm điều khiển từ xa, lập trình trước, hoặc thông qua các giao diện người-máy như cảm biến chuyển động.
- Với sự phát triển của trí tuệ nhân tạo và học máy, nhiều cánh tay robot hiện đại có khả năng học hỏi và tự động cải thiện khả năng làm việc của mình.

1.1.4. Ứng dụng:

- **Công nghiệp:** Trong sản xuất ô tô, đồ điện tử, và nhiều lĩnh vực khác.
- **Y tế:** Hỗ trợ phẫu thuật, phục hồi chức năng.
- **Nghiên cứu:** Thí nghiệm trong lĩnh vực sinh học, vật lý.
- **Giải trí:** Trong các trò chơi điện tử và phim ảnh.

Cánh tay robot đã trở thành một phần quan trọng trong nhiều quy trình sản xuất và nghiên cứu hiện đại, giúp tăng cường hiệu quả làm việc, giảm thiểu rủi ro cho con người, và thực hiện những nhiệm vụ phức tạp mà con người khó có thể làm.

Nhìn chung đây là bước mở đầu để hiểu biết thêm về các cơ cấu trong cánh tay robot công nghiệp.

1.2. Mục tiêu cần đạt của bài thuyết trình:

Mô hình robot này bao gồm 6 khớp là các servo hoạt động độc lập, sử dụng phần mềm chuyên dụng để có khả năng kiểm soát các góc độ cho từng servo dựa vào việc cấp xung. Trong quá trình đo đạc lấy dữ liệu, chúng ta có thể suy ra được tọa độ của điểm P cần gấp, số xung tương ứng với các góc độ servo, kích thước của từng khớp. Điều này giúp chúng ta có thể dễ dàng lập trình và dùng các thông số đó để tính toán động học thuận, động học nghịch để dễ dàng áp dụng vào công nghiệp trong quá trình làm việc tương lai

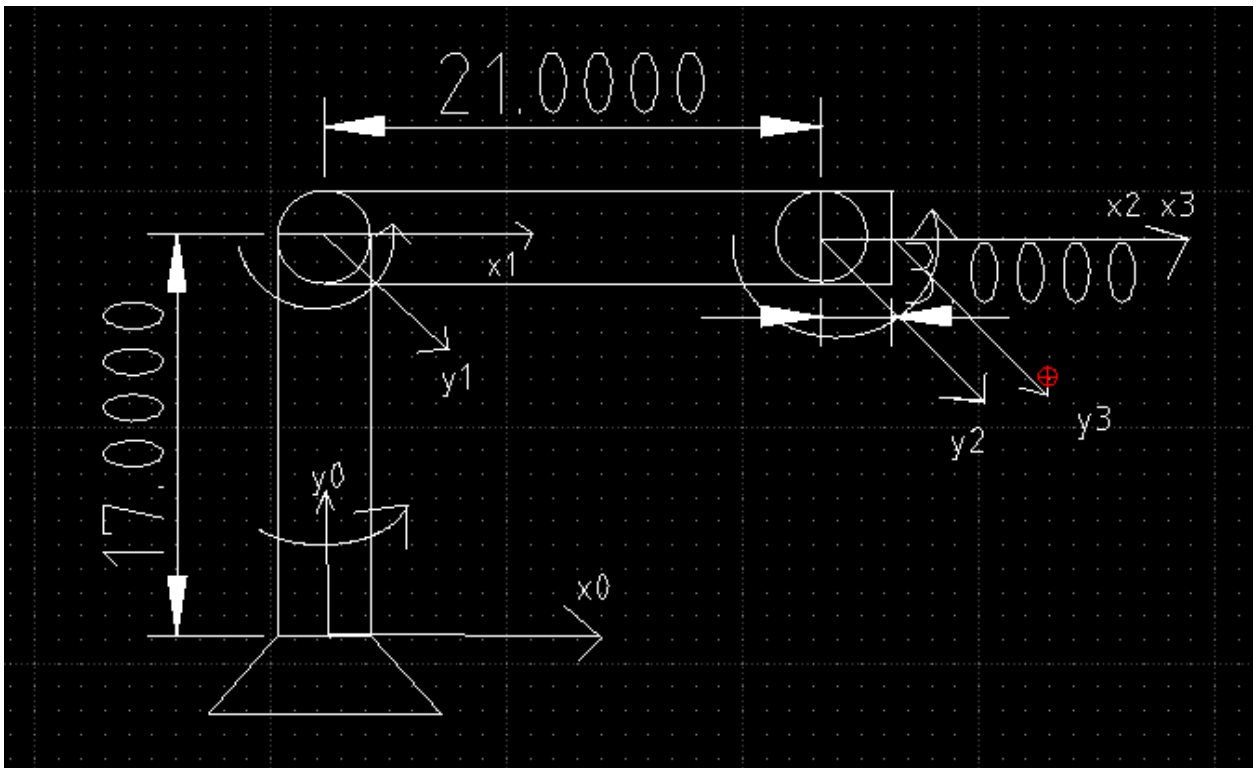
Thông qua việc tiếp xúc với mô hình Robot 6 khớp nhóm chúng em hiểu được cấu tạo, nguyên lý hoạt động của cả quá trình, cũng như từ các công cụ có sẵn để xây dựng lên một quá trình học tập và tìm hiểu cũng như nâng cao khả năng học tập

CHƯƠNG 2. NGUYÊN LÝ HOẠT ĐỘNG

2.1. Kết cấu cơ khí

Mô hình Robot này bao gồm các bậc tự do, các khâu, các khớp:

- **Bậc tự do** 6 (thực tế thường là **5 khớp quay + 1 kìm/gripper** mở–đóng; nhiều kit gọi chung là 6 DOF).
- **Số khớp**: 6 khớp.
- **Số khâu**: 7 khâu (tính cả đế và khâu công tác).



Sơ đồ lập các hệ tọa độ cho từng khâu

Chiều dài các khâu

$$L1 = 17\text{cm}$$

$$L2 = 21\text{cm}$$

$$L3 = 3\text{cm}$$

Bước tiến của từng servo

Để có thể dễ dàng điều khiển và lập trình cánh tay robot, chúng ta phải biết cách để có thể điều khiển servo đến góc độ mà chúng ta mong muốn. Trong trường hợp này, có phần mềm để lập trình số xung để điều khiển góc, bước tiến của từng servo với cơ cấu chính là phần tối quan trọng để có thể biết chính xác được số xung tương ứng với góc của khớp.

Bước tiến xung mỗi khi khớp thay đổi 10 độ góc:

Servo 1	Servo 2	Servo 3	Servo 4	Servo 5	Servo 6
72.222	50	116.7	105.55	111.11	105.55

Xác định tọa độ điểm P(điểm cuối của cánh tay robot) bằng phương pháp động học thuận qua 6 trường hợp

TH1: Khâu 1 giữ nguyên, khâu 3 giữ nguyên.

TH2: Khâu 1 giữ nguyên, khâu 3 quay lên 90 độ.

TH3: Khâu 1 giữ nguyên, Khâu 3 quay xuống 90 độ.

TH4: Khâu 1 quay ngang 90 độ, khâu 3 giữ nguyên.

TH5: Khâu 1 quay ngang 90 độ, khâu 3 quay lên 90 độ.

TH6: Khâu 1 quay ngang 90 độ, khâu 3 quay xuống 90 độ.

2.2. Điện – Điện tử

2.2.1. Động cơ Servo

- DS-S006M

DS-S006M là một servo bánh răng kim loại, kỹ thuật số, siêu nhỏ, thường được gọi là servo 9g Đây là một servo mô-men xoắn cao, phù hợp để nâng cấp cho các servo mini tiêu chuẩn trên xe mô hình tỷ lệ 1/16. Nó hoạt động ở mức điện áp từ 4,8V đến 6V và có mô-men xoắn dừng >1,3kgf.cm ở 4,8V và >1,5kgf.cm ở 6,0V. @

Thông số kỹ thuật chính:

- Điện áp hoạt động: 4,8V-6V DC
- Tốc độ không tải: <0,13 giây/60° ở 4,8V, <0,11 giây/60° ở 6,0V
- Mô-men xoắn dừng: >1,3kgf.cm ở 4,8V, >1,5kg f.cm ở 6,0V
- Phạm vi độ rộng xung: 500-2500uS
- Góc di chuyển vận hành: 90° +10°

- Góc giới hạn cơ học: 210°
- Trọng lượng: $13,5g \pm 0,5g$ (bao gồm dây)
- Kích thước: $27,3 \times 12 \times 23mm$
- Vật liệu bánh răng: Kim loại
- Loại động cơ: Không lõi
- Thương hiệu: DS Power
- Mã sản phẩm: DS-S006M

- **Servo KS-3518**

Servo KS-3518 là một động cơ servo kỹ thuật số với mô-men xoắn dừng $20kg\cdot cm$ ở $6,6V$ và tốc độ $0,16$ giây ở $6,6V$. Sản phẩm có thiết kế chống nước, hệ thống bánh răng kim loại và ổ bi kép giúp tăng cường độ bền và hiệu suất. Sản phẩm hoạt động trong dải điện áp từ $4,8V$ đến $6,6V$ và phù hợp với nhiều ứng dụng khác nhau như robot điều khiển từ xa, ô tô và tàu thuyền.

Thông số kỹ thuật chính:

- Mô-men xoắn: $20\ kg\cdot cm$ (ở $6,6V$)
- Tốc độ: $0,16$ giây (ở $6,6V$)
- Dải điện áp: $4,8V$ đến $6,6V$
- Loại bánh răng: Kim loại
- Loại vòng bi: Vòng bi kép
- Chống thấm nước: Có
- Trọng lượng: Khoảng $60g$
- Góc hoạt động: $90^\circ \times 2$ (tổng cộng 180°)
- Loại động cơ: Động cơ DC
- Tần số làm việc: $500us/2500hz$ (theo một danh sách, AliExpress)

2.2.2. Điều khiển Robot

Mô hình robot này bao gồm 6 khớp là các servo hoạt động độc lập, sử dụng phần mềm chuyên dụng để có khả năng kiểm soát các góc độ cho từng servo dựa vào việc cấp xung.

Ngoài ra cần tính toán các khâu để đạt được vị trí mong muốn.

TH 1: Khâu 1 giữ nguyên khâu 3 giữ nguyên.

Bảng D-H

Khâu	a	α	d	θ_i
1	0	90	17	0
2	21	0	0	0
3	3	0	0	0

Tính toán 3 ma trận:

$$T1 = \begin{bmatrix} \cos 0^\circ & -\sin 0^\circ \times \cos 90^\circ & \sin 0^\circ \times \sin 90^\circ & 0 \cos 0^\circ \\ \sin 0^\circ & \cos 0^\circ \times \cos 90^\circ & -\cos 0^\circ \times \sin 90^\circ & 0 \sin 0^\circ \\ 0 & \sin 90^\circ & \cos 90^\circ & 17 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T2 = \begin{bmatrix} \cos 0^\circ & -\sin 0^\circ \times \cos 0^\circ & \sin 0^\circ \times \sin 0^\circ & 21 \cos 0^\circ \\ \sin 0^\circ & \cos 0^\circ \times \cos 0^\circ & -\cos 0^\circ \times \sin 0^\circ & 21 \sin 0^\circ \\ 0 & \sin 0^\circ & \cos 0^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T3 = \begin{bmatrix} \cos 0^\circ & -\sin 0^\circ \times \cos 0^\circ & \sin 0^\circ \times \sin 0^\circ & 3 \cos 0^\circ \\ \sin 0^\circ & \cos 0^\circ \times \cos 0^\circ & -\cos 0^\circ \times \sin 0^\circ & 3 \sin 0^\circ \\ 0 & \sin 0^\circ & \cos 0^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Nhân 3 ma trận: $T = T1.T2.T3$

$$T = \begin{bmatrix} 1 & 0 & 0 & 24 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & -2 & 17 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vì vậy tọa độ điểm P là P (24;0;17), dựa vào phương pháp hình học ,

đây là kết quả chính xác.

TH2: Khâu 1 giữ nguyên, khâu 3 quay lên 90 độ:

Bảng D-H

Khâu	a	α	d	θ
1	0	90	17	0
2	21	0	0	0
3	3	0	0	90

Tính toán 3 ma trận:

$$T1 = \begin{bmatrix} \cos 0^\circ & -\sin 0^\circ \times \cos 90^\circ & \sin 0^\circ \times \sin 90^\circ & 0 \cos 0^\circ \\ \sin 0^\circ & \cos 0^\circ \times \cos 90^\circ & -\cos 0^\circ \times \sin 90^\circ & 0 \sin 0^\circ \\ 0 & \sin 90^\circ & \cos 90^\circ & 17 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T2 = \begin{bmatrix} \cos 0^\circ & -\sin 0^\circ \times \cos 0^\circ & \sin 0^\circ \times \sin 0^\circ & 21 \cos 0^\circ \\ \sin 0^\circ & \cos 0^\circ \times \cos 0^\circ & -\cos 0^\circ \times \sin 0^\circ & 21 \sin 0^\circ \\ 0 & \sin 0^\circ & \cos 0^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T3 = \begin{bmatrix} \cos 90^\circ & -\sin 90^\circ \times \cos 0^\circ & \sin 90^\circ \times \sin 0^\circ & 3 \cos 90^\circ \\ \sin 90^\circ & \cos 90^\circ \times \cos 0^\circ & -\cos 90^\circ \times \sin 0^\circ & 3 \sin 90^\circ \\ 0 & \sin 0^\circ & \cos 0^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Nhân 3 ma trận: $T = T1.T2.T3$

$$T = \begin{bmatrix} 0 & -1 & 0 & 21 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & -1 & 20 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vì vậy tọa độ điểm P là P (21;0;20), dựa vào phương pháp hình học, đây là kết quả chính xác.

TH3: Khâu 1 giữ nguyên, khâu 3 quay xuống 90 độ.

Bảng D-H

Khâu	a	α	d	θ
1	0	90	17	0
2	21	0	0	0
3	3	0	0	-90

Tính toán 3 ma trận:

$$T1 = \begin{bmatrix} \cos 0^\circ & -\sin 0^\circ \times \cos 90^\circ & \sin 0^\circ \times \sin 90^\circ & 0 \cos 0^\circ \\ \sin 0^\circ & \cos 0^\circ \times \cos 90^\circ & -\cos 0^\circ \times \sin 90^\circ & 0 \sin 0^\circ \\ 0 & \sin 90^\circ & \cos 90^\circ & 17 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T2 = \begin{bmatrix} \cos 0^\circ & -\sin 0^\circ \times \cos 0^\circ & \sin 0^\circ \times \sin 0^\circ & 21 \cos 0^\circ \\ \sin 0^\circ & \cos 0^\circ \times \cos 0^\circ & -\cos 0^\circ \times \sin 0^\circ & 21 \sin 0^\circ \\ 0 & \sin 0^\circ & \cos 0^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T3 = \begin{bmatrix} \cos -90^\circ & -\sin -90^\circ \times \cos 0^\circ & \sin -90^\circ \times \sin 0^\circ & 3 \cos -90^\circ \\ \sin -90^\circ & \cos -90^\circ \times \cos 0^\circ & -\cos -90^\circ \times \sin 0^\circ & 3 \sin -90^\circ \\ 0 & \sin 0^\circ & \cos 0^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Nhân 3 ma trận: $T = T1.T2.T3$

$$T = \begin{bmatrix} 0 & 1 & 0 & 21 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & -1 & 14 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vì vậy tọa độ điểm P là P (21;0;14), dựa vào phương pháp hình

học, đây là kết quả chính xác.

TH4: Khâu 1 quay ngang 90 độ, khâu 3 giữ nguyên.

Bảng D-H

Khâu	a	α	d	θ
1	0	90	17	90
2	21	0	0	0
3	3	0	0	0

Tính toán 3 ma trận:

$$T1 = \begin{bmatrix} \cos 90^\circ & -\sin 90^\circ \times \cos 90^\circ & \sin 90^\circ \times \sin 90^\circ & 0 \cos 90^\circ \\ \sin 90^\circ & \cos 90^\circ \times \cos 90^\circ & -\cos 90^\circ \times \sin 90^\circ & 0 \sin 90^\circ \\ 0 & \sin 90^\circ & \cos 90^\circ & 17 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T2 = \begin{bmatrix} \cos 0^\circ & -\sin 0^\circ \times \cos 0^\circ & \sin 0^\circ \times \sin 0^\circ & 21 \cos 0^\circ \\ \sin 0^\circ & \cos 0^\circ \times \cos 0^\circ & -\cos 0^\circ \times \sin 0^\circ & 21 \sin 0^\circ \\ 0 & \sin 0^\circ & \cos 0^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T3 = \begin{bmatrix} \cos 0^\circ & -\sin 0^\circ \times \cos 0^\circ & \sin 0^\circ \times \sin 0^\circ & 3 \cos 0^\circ \\ \sin 0^\circ & \cos 0^\circ \times \cos 0^\circ & -\cos 0^\circ \times \sin 0^\circ & 3 \sin 0^\circ \\ 0 & \sin 0^\circ & \cos 0^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Nhân 3 ma trận: $T = T1.T2.T3$

$$T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 24 \\ 0 & 1 & 0 & 17 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vì vậy tọa độ điểm P là P (0;24;17), dựa vào phương pháp hình học , đây

là kết quả chính xác.

TH5: Khâu 1 quay ngang 90 độ, khâu 3 quay lên 90 độ.

Bảng D-H

Khâu	a	α	d	θ
1	0	90	17	90

2	21	0	0	0
3	3	0	0	90

Tính toán 3 ma trận:

$$T1 = \begin{bmatrix} \cos 90^\circ & -\sin 90^\circ \times \cos 90^\circ & \sin 90^\circ \times \sin 90^\circ & 0 \cos 90^\circ \\ \sin 90^\circ & \cos 90^\circ \times \cos 90^\circ & -\cos 90^\circ \times \sin 90^\circ & 0 \sin 90^\circ \\ 0 & \sin 90^\circ & \cos 90^\circ & 17 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T2 = \begin{bmatrix} \cos 0^\circ & -\sin 0^\circ \times \cos 0^\circ & \sin 0^\circ \times \sin 0^\circ & 21 \cos 0^\circ \\ \sin 0^\circ & \cos 0^\circ \times \cos 0^\circ & -\cos 0^\circ \times \sin 0^\circ & 21 \sin 0^\circ \\ 0 & \sin 0^\circ & \cos 0^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T3 = \begin{bmatrix} \cos 90^\circ & -\sin 90^\circ \times \cos 0^\circ & \sin 90^\circ \times \sin 0^\circ & 3 \cos 90^\circ \\ \sin 90^\circ & \cos 90^\circ \times \cos 0^\circ & -\cos 90^\circ \times \sin 0^\circ & 3 \sin 90^\circ \\ 0 & \sin 0^\circ & \cos 0^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Nhân 3 ma trận: $T = T1.T2.T3$

$$T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 21 \\ 1 & 0 & 0 & 20 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vì vậy tọa độ điểm P là P (0;21;20), dựa vào phương pháp hình học ,

đây là kết quả chính xác.

TH6: Khâu 1 quay ngang 90 độ, khâu 3 quay xuống 90 độ.

Bảng D-H

Khâu	a	α	d	θ
1	0	90	17	90
2	21	0	0	0
3	3	0	0	-90

Tính toán 3 ma trận:

$$T1 = \begin{bmatrix} \cos 90^\circ & -\sin 90^\circ \times \cos 90^\circ & \sin 90^\circ \times \sin 90^\circ & 0 \cos 90^\circ \\ \sin 90^\circ & \cos 90^\circ \times \cos 90^\circ & -\cos 90^\circ \times \sin 90^\circ & 0 \sin 90^\circ \\ 0 & \sin 90^\circ & \cos 90^\circ & 17 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T2 = \begin{bmatrix} \cos 0^\circ & -\sin 0^\circ \times \cos 0^\circ & \sin 0^\circ \times \sin 0^\circ & 21 \cos 0^\circ \\ \sin 0^\circ & \cos 0^\circ \times \cos 0^\circ & -\cos 0^\circ \times \sin 0^\circ & 21 \sin 0^\circ \\ 0 & \sin 0^\circ & \cos 0^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T3 = \begin{bmatrix} \cos -90^\circ & -\sin -90^\circ \times \cos 0^\circ & \sin -90^\circ \times \sin 0^\circ & 3 \cos -90^\circ \\ \sin -90^\circ & \cos -90^\circ \times \cos 0^\circ & -\cos -90^\circ \times \sin 0^\circ & 3 \sin -90^\circ \\ 0 & \sin 0^\circ & \cos 0^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

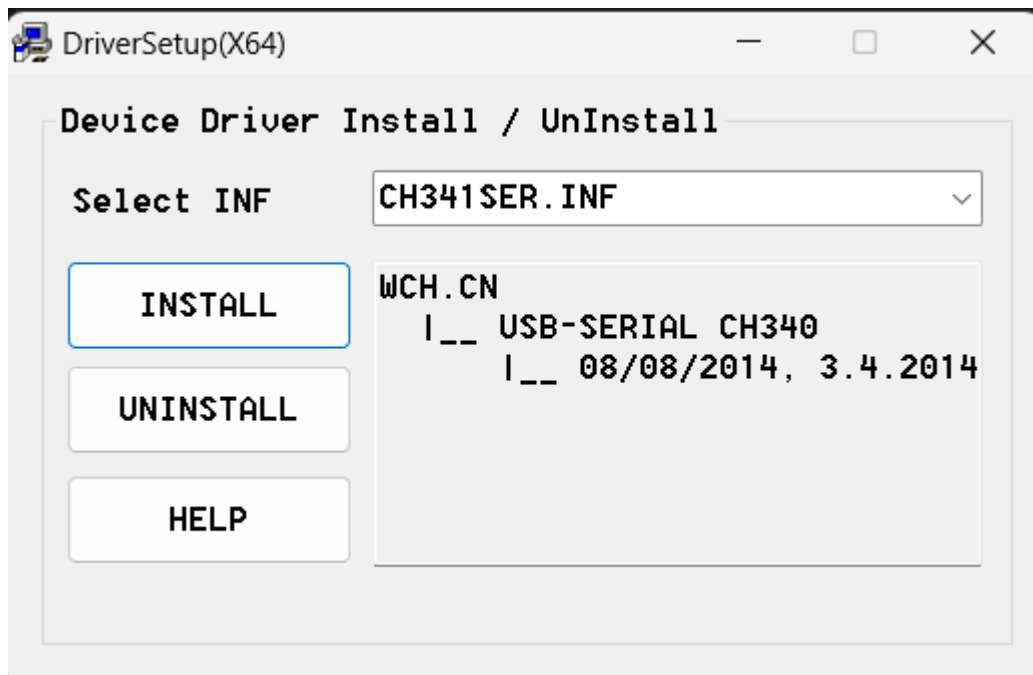
Nhân 3 ma trận: $T = T1.T2.T3$

$$T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 21 \\ -1 & 0 & 0 & 14 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vì vậy tọa độ điểm P là P (0;21;14), dựa vào phương pháp hình học ,

đây là kết quả chính xác.

2.3. Truyền thông



Sử dụng Driver CH341SER để nhận dữ liệu driver của Robot dành cho việc kết nối

Tổng quan về driver CH341SER

CH341SER là trình điều khiển (driver) cho chip CH340/CH341 của hãng WCH, dùng để chuyển đổi USB sang cổng Serial (UART).

Khi cài driver này, máy tính sẽ nhận thiết bị dùng chip CH340/CH341 như một cổng COM ảo, cho phép truyền dữ liệu hoặc nạp chương trình.

- Ứng dụng phổ biến
 - Bo Arduino giá rẻ
 - Mạch nạp vi điều khiển
 - Module GPS, thiết bị đo, máy in 3D...
- Chức năng chính
 - Kết nối thiết bị Serial qua cổng USB
 - Tạo COM ảo để phần mềm giao tiếp dễ dàng.

CHƯƠNG 3. THIẾT KẾ GIAO DIỆN ĐIỀU KHIỂN

3.1. Thiết kế Giao diện điều khiển

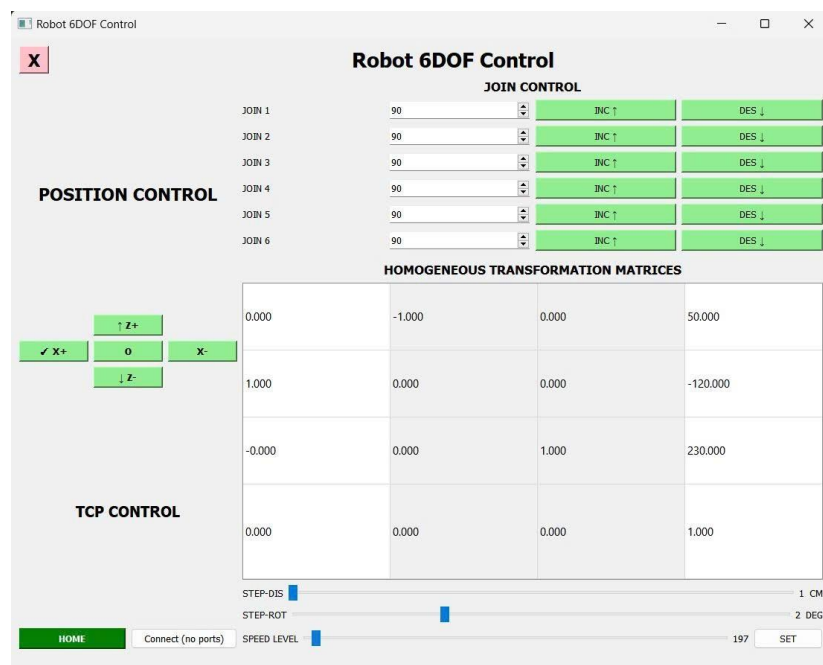
3.1.1. Phần mềm Qt Designer

Tổng quan về Qt Designer

Qt Designer là công cụ trực quan đi kèm bộ Qt, cho phép thiết kế giao diện người dùng (GUI) bằng cách kéo-thả mà không cần viết mã giao diện thủ công.

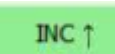
- **Chức năng chính:** tạo cửa sổ, hộp thoại, nút bấm, menu, form nhập liệu... và sắp xếp chúng bằng layout.
- **Định dạng lưu:** .ui (XML) – mô tả toàn bộ bố cục và thuộc tính giao diện.
- Tích hợp lập trình:
 - Chuyển .ui → .py (PyQt/PySide) bằng pyuic.
 - Hoặc load trực tiếp .ui trong code.
- **Signal–Slot:** cơ chế liên kết sự kiện (như click nút) với hàm xử lý logic.
- **Ứng dụng:** phát triển nhanh các phần mềm desktop đa nền tảng, giảm thời gian viết mã giao diện.


3.1.2. Thiết kế giao diện



Giải thích giao diện:


JOIN  : Vị trí bậc mà Robot di chuyển


INC  : Tăng góc độ của Servo điều khiển


DES  : Giảm góc độ của Servo điều khiển


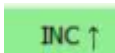

 : Giá trị góc của Servo điều khiển

 : Đưa Robot trở về vị trí Ban đầu

 : Kết nối với cổng truyền thông

STEP-DIS  : Setup bước của cánh tay theo Cm

STEP-ROT  : Setup bước của cánh tay theo góc độ

SPEED LEVEL  : Tốc độ gia tăng từng bước của Robot Khi bấm vào INC  hoặc DES 

HOMOGENEOUS TRANSFORMATION MATRICES			
0.000	-1.000	0.000	50.000
1.000	0.000	0.000	-120.000
-0.000	0.000	1.000	230.000
0.000	0.000	0.000	1.000

Bảng giá trị D - H của Robot

3.2. Lập trình thiết kế giao diện trên VS Code

VS Code (Visual Studio Code) là trình soạn thảo mã nguồn miễn phí, đa nền tảng do Microsoft phát triển.

Đặc điểm chính:

- Hỗ trợ nhiều ngôn ngữ lập trình (Python, C/C++, Java, JavaScript, PHP, v.v.)
- Hệ thống extension phong phú: cài thêm để hỗ trợ Qt, Arduino, Git, Docker...
- Tích hợp Terminal và Git ngay trong môi trường làm việc
- Có Debugger mạnh mẽ, hỗ trợ gỡ lỗi nhiều ngôn ngữ
- Giao diện tùy biến, hỗ trợ theme và keymapd

❖ **Lập trình thiết kế giao diện trên VS Code qua Qt Designer**

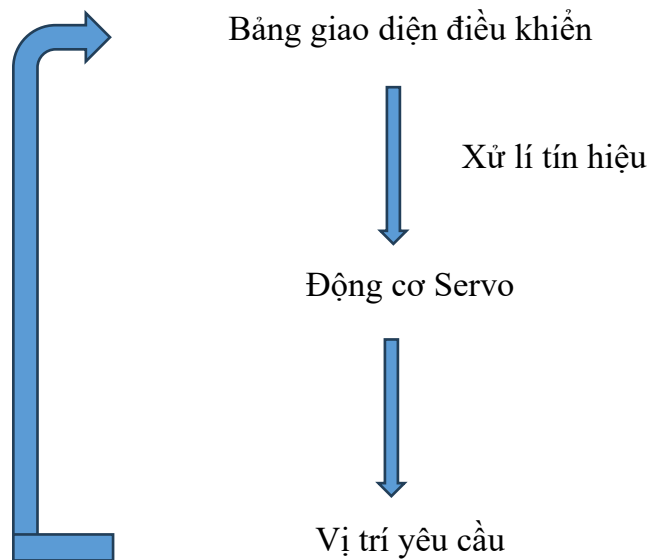
- ✓ Mở Qt Designer, chọn loại form (Main Window, Dialog, Widget...).
- ✓ Kéo-thả widget vào cửa sổ thiết kế, chỉnh thuộc tính trong Property Editor.
- ✓ Sắp xếp giao diện bằng layout (Vertical, Horizontal, Grid...).
- ✓ Lưu giao diện thành file .ui.
- ✓ Tích hợp vào dự án trong VS Code

CHƯƠNG 4. SƠ ĐỒ GIẢI THUẬT CỦA ĐIỀU KHIỂN ROBOT

4.1. Sơ đồ giải thuật:

- Mô tả sơ đồ giải thuật:
 - Nhận lệnh điều khiển từ giao diện được thiết kế trước đó
 - Xử lý tín hiệu nhận được từ giao diện
 - Động cơ hoạt động theo yêu cầu
 - Trả kết quả và hiển thị lại trên bảng điều khiển

4.2. Lưu đồ giải thuật:



CHƯƠNG 5. KẾT QUẢ

Sau quá trình thiết kế, lắp ráp và kiểm thử, đã đạt được các kết quả sau:

1. Vận hành cơ cấu chính xác và ổn định:

- Robot thực hiện đầy đủ 6 bậc tự do (3 bậc tịnh tiến + 3 bậc quay) một cách linh hoạt, giúp mô phỏng gần giống thao tác tay người.
- Góc quay và tầm với của các khớp phù hợp với các yêu cầu tác vụ thông thường như gấp, di chuyển, lắp ráp hoặc sắp xếp.

2. Hiệu suất điều khiển cao:

- Hệ thống điều khiển sử dụng vi điều khiển cho khả năng phản hồi tốt, sai số vị trí đầu cuối (end-effector) đo được nhỏ hơn ± 2 mm.
- Có thể điều khiển thông qua giao diện phần mềm (máy tính) hoặc tay cầm điều khiển trực tiếp.

3. Khả năng lặp lại tốt

- Robot cho độ lặp lại ổn định sau nhiều lần thực hiện một hành trình nhất định, sai số lặp lại đo được $< \pm 1.5$ mm.

4. Tải trọng và tốc độ làm việc:

- Tải trọng tối đa ở đầu công tác đạt $\sim 500\text{g} - 1\text{kg}$ (tùy cấu hình servo).
- Tốc độ hoạt động phù hợp với các tác vụ gấp – đặt cơ bản trong giáo dục và nghiên cứu.

5. Tích hợp mô phỏng và lập trình ngoại tuyến:

- Mô phỏng chuyển động trên phần mềm (ví dụ: MATLAB, ROS, hoặc SolidWorks) cho thấy độ tương thích cao giữa mô hình lý thuyết và thực tế.
- Hỗ trợ lập trình lộ trình chuyển động theo tọa độ không gian hoặc thông qua ghi nhớ thao tác (teaching).

6. Khả năng mở rộng và tích hợp:

- Cho phép dễ dàng tích hợp với cảm biến lực, thị giác máy tính hoặc băng tải để phát triển thành một hệ thống tự động hóa hoàn chỉnh.

CHƯƠNG 6. KẾT LUẬN

Sau quá trình nghiên cứu, thiết kế và thử nghiệm, đã chứng minh được khả năng hoạt động linh hoạt, chính xác và ổn định trong các tác vụ mô phỏng thao tác của con người. Hệ thống điều khiển cho phép robot thực hiện các chuyển động không gian phức tạp nhờ cấu trúc 6 bậc tự do, đồng thời dễ dàng tích hợp với các hệ thống thị giác máy tính hoặc cảm biến bên ngoài để tăng độ chính xác và khả năng tự động hóa.

Robot được đánh giá cao ở các khía cạnh:

- **Thiết kế cơ khí tối ưu**, nhỏ gọn nhưng vẫn đảm bảo độ cứng vững và khả năng tải trọng phù hợp với các ứng dụng thực tiễn.
- **Hệ thống điều khiển hiệu quả**, có thể vận hành ở chế độ điều khiển tay hoặc tự động theo lập trình.
- **Tính linh hoạt cao**, thích hợp cho các ứng dụng trong giáo dục, nghiên cứu robot học, và một số quy trình sản xuất công nghiệp nhẹ.

Tuy nhiên, để đạt hiệu quả cao hơn trong môi trường thực tế, robot AMZ_R3 có thể được nâng cấp thêm về khả năng cảm biến lực – mô-men, tăng tốc độ phản hồi hệ thống và tích hợp AI để cải thiện khả năng học hỏi và thích nghi với môi trường làm việc.

Mục lục

Chương trình Giao diện

```
# -*- coding: utf-8 -*-
# Form implementation generated from reading ui file 'robot_control.ui'
#
# Created by: PyQt5 UI code generator 5.15.10
#
# WARNING: Any manual changes made to this file will be lost when pyuic5 is
# run again. Do not edit this file unless you know what you are doing.
from PyQt5 import QtCore, QtGui, QtWidgets
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(1102, 857)
        MainWindow.setSizeIncrement(QtCore.QSize(0, 0))
        MainWindow.setBaseSize(QtCore.QSize(4, 4))
        MainWindow.setMouseTracking(False)
        MainWindow.setLayoutDirection(QtCore.Qt.LeftToRight)
        MainWindow.setAutoFillBackground(False)
        MainWindow.setDocumentMode(False)
        MainWindow.setDockNestingEnabled(False)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.verticalLayout = QtWidgets.QVBoxLayout(self.centralwidget)
        self.verticalLayout.setObjectName("verticalLayout")
        self.topLayout = QtWidgets.QHBoxLayout()
        self.topLayout.setObjectName("topLayout")
        self.btn_close = QtWidgets.QPushButton(self.centralwidget)
        self.btn_close.setMaximumSize(QtCore.QSize(40, 40))
        self.btn_close.setStyleSheet("background-color: pink; font-weight: bold; font-size: 14pt;")
        self.btn_close.setObjectName("btn_close")
        self.topLayout.addWidget(self.btn_close)
        spacerItem = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Maximum, QtWidgets.QSizePolicy.Minimum)
        self.topLayout.addItem(spacerItem)
        self.titleLabel = QtWidgets.QLabel(self.centralwidget)
        self.titleLabel.setStyleSheet("font-size: 16pt; font-weight: bold;")
```

```

self.titleLabel.setAlignment(QtCore.Qt.AlignCenter)
self.titleLabel.setObjectName("titleLabel")
self.topLayout.addWidget(self.titleLabel)
self.verticalLayout.addLayout(self.topLayout)
self.mainContent = QtWidgets.QHBoxLayout()
self.mainContent.setObjectName("mainContent")
self.leftColumn = QtWidgets.QVBoxLayout()
self.leftColumn.setObjectName("leftColumn")
self.lbl_position = QtWidgets.QLabel(self.centralwidget)
font = QtGui.QFont()
font.setPointSize(14)
font.setBold(True)
font.setWeight(75)
self.lbl_position.setFont(font)
self.lbl_position.setStyleSheet("font-weight: bold;")
self.lbl_position.setAlignment(QtCore.Qt.AlignCenter)
self.lbl_position.setObjectName("lbl_position")
self.leftColumn.addWidget(self.lbl_position)
self.grid_position = QtWidgets.QGridLayout()
self.grid_position.setObjectName("grid_position")
self.btn_z_plus = QtWidgets.QPushButton(self.centralwidget)
self.btn_z_plus.setStyleSheet("background-color: lightgreen; font-weight: bold;")
self.btn_z_plus.setObjectName("btn_z_plus")
self.grid_position.addWidget(self.btn_z_plus, 0, 1, 1, 1)
self.btn_o = QtWidgets.QPushButton(self.centralwidget)
self.btn_o.setStyleSheet("background-color: lightgreen; font-weight: bold;")
self.btn_o.setObjectName("btn_o")
self.grid_position.addWidget(self.btn_o, 1, 1, 1, 1)
self.btn_z_minus = QtWidgets.QPushButton(self.centralwidget)
self.btn_z_minus.setStyleSheet("background-color: lightgreen; font-weight: bold;")
self.btn_z_minus.setObjectName("btn_z_minus")
self.grid_position.addWidget(self.btn_z_minus, 2, 1, 1, 1)
self.btn_x_plus = QtWidgets.QPushButton(self.centralwidget)
self.btn_x_plus.setStyleSheet("background-color: lightgreen; font-weight: bold;")
self.btn_x_plus.setObjectName("btn_x_plus")
self.grid_position.addWidget(self.btn_x_plus, 1, 0, 1, 1)
self.btn_x_minus = QtWidgets.QPushButton(self.centralwidget)
self.btn_x_minus.setStyleSheet("background-color: lightgreen; font-weight: bold;")

```

```

self.btn_x_minus.setObjectName("btn_x_minus")
self.grid_position.addWidget(self.btn_x_minus, 1, 2, 1, 1)
self.leftColumn.addLayout(self.grid_position)
self.lbl_tcp = QtWidgets.QLabel(self.centralwidget)
font = QtGui.QFont()
font.setPointSize(12)
font.setBold(True)
font.setWeight(75)
self.lbl_tcp.setFont(font)
self.lbl_tcp.setStyleSheet("font-weight: bold; margin-top: 8px;")
self.lbl_tcp.setAlignment(QtCore.Qt.AlignCenter)
self.lbl_tcp.setObjectName("lbl_tcp")
self.leftColumn.addWidget(self.lbl_tcp)
self.grip_home = QtWidgets.QHBoxLayout()
self.grip_home.setObjectName("grip_home")
self.btn_home_small = QtWidgets.QPushButton(self.centralwidget)
self.btn_home_small.setStyleSheet("background-color: green; color: white; font-weight: bold;")
self.btn_home_small.setObjectName("btn_home_small")
self.grip_home.addWidget(self.btn_home_small)
self.btn_connect = QtWidgets.QPushButton(self.centralwidget)
self.btn_connect.setObjectName("btn_connect")
self.grip_home.addWidget(self.btn_connect)
self.leftColumn.addLayout(self.grip_home)
self.mainContent.addLayout(self.leftColumn)
self.rightColumn = QtWidgets.QVBoxLayout()
self.rightColumn.setObjectName("rightColumn")
self.lbl_join_big = QtWidgets.QLabel(self.centralwidget)
font = QtGui.QFont()
font.setPointSize(10)
font.setBold(True)
font.setWeight(75)
self.lbl_join_big.setFont(font)
self.lbl_join_big.setStyleSheet("font-weight: bold;")
self.lbl_join_big.setAlignment(QtCore.Qt.AlignCenter)
self.lbl_join_big.setObjectName("lbl_join_big")
self.rightColumn.addWidget(self.lbl_join_big)
self.grid_join_full = QtWidgets.QGridLayout()
self.grid_join_full.setObjectName("grid_join_full")

```

```

self.btn_inc_J3 = QtWidgets.QPushButton(self.centralwidget)
self.btn_inc_J3.setStyleSheet("background-color: lightgreen;")
self.btn_inc_J3.setObjectName("btn_inc_J3")
self.grid_join_full.addWidget(self.btn_inc_J3, 2, 2, 1, 1)
self.btn_des_J3 = QtWidgets.QPushButton(self.centralwidget)
self.btn_des_J3.setStyleSheet("background-color: lightgreen;")
self.btn_des_J3.setObjectName("btn_des_J3")
self.grid_join_full.addWidget(self.btn_des_J3, 2, 3, 1, 1)
self.lbl_J3 = QtWidgets.QLabel(self.centralwidget)
self.lbl_J3.setObjectName("lbl_J3")
self.grid_join_full.addWidget(self.lbl_J3, 2, 0, 1, 1)
self.spin_J3 = QtWidgets.QSpinBox(self.centralwidget)
self.spin_J3.setMinimum(-360)
self.spin_J3.setMaximum(360)
self.spin_J3.setProperty("value", 89)
self.spin_J3.setObjectName("spin_J3")
self.grid_join_full.addWidget(self.spin_J3, 2, 1, 1, 1)
self.spin_J4 = QtWidgets.QSpinBox(self.centralwidget)
self.spin_J4.setMinimum(-360)
self.spin_J4.setMaximum(360)
self.spin_J4.setProperty("value", 89)
self.spin_J4.setObjectName("spin_J4")
self.grid_join_full.addWidget(self.spin_J4, 3, 1, 1, 1)
self.btn_des_J4 = QtWidgets.QPushButton(self.centralwidget)
self.btn_des_J4.setStyleSheet("background-color: lightgreen;")
self.btn_des_J4.setObjectName("btn_des_J4")
self.grid_join_full.addWidget(self.btn_des_J4, 3, 3, 1, 1)
self.btn_inc_J4 = QtWidgets.QPushButton(self.centralwidget)
self.btn_inc_J4.setStyleSheet("background-color: lightgreen;")
self.btn_inc_J4.setObjectName("btn_inc_J4")
self.grid_join_full.addWidget(self.btn_inc_J4, 3, 2, 1, 1)
self.lbl_J4 = QtWidgets.QLabel(self.centralwidget)
self.lbl_J4.setObjectName("lbl_J4")
self.grid_join_full.addWidget(self.lbl_J4, 3, 0, 1, 1)
self.lbl_J5 = QtWidgets.QLabel(self.centralwidget)
self.lbl_J5.setObjectName("lbl_J5")
self.grid_join_full.addWidget(self.lbl_J5, 4, 0, 1, 1)
self.btn_des_J5 = QtWidgets.QPushButton(self.centralwidget)

```



```

self.btn_des_J5.setStyleSheet("background-color: lightgreen;")
self.btn_des_J5.setObjectName("btn_des_J5")
self.grid_join_full.addWidget(self.btn_des_J5, 4, 3, 1, 1)
self.btn_inc_J5 = QtWidgets.QPushButton(self.centralwidget)
self.btn_inc_J5.setStyleSheet("background-color: lightgreen;")
self.btn_inc_J5.setObjectName("btn_inc_J5")
self.grid_join_full.addWidget(self.btn_inc_J5, 4, 2, 1, 1)
self.spin_J5 = QtWidgets.QSpinBox(self.centralwidget)
self.spin_J5.setMinimum(-360)
self.spin_J5.setMaximum(360)
self.spin_J5.setProperty("value", 89)
self.spin_J5.setObjectName("spin_J5")
self.grid_join_full.addWidget(self.spin_J5, 4, 1, 1, 1)
self.lbl_J6 = QtWidgets.QLabel(self.centralwidget)
self.lbl_J6.setObjectName("lbl_J6")
self.grid_join_full.addWidget(self.lbl_J6, 5, 0, 1, 1)
self.lbl_J1 = QtWidgets.QLabel(self.centralwidget)
self.lbl_J1.setEnabled(True)
self.lbl_J1.setMouseTracking(False)
self.lbl_J1.setTabletTracking(False)
self.lbl_J1.setFocusPolicy(QtCore.Qt.StrongFocus)
self.lbl_J1.setContextMenuPolicy(QtCore.Qt.CustomContextMenu)
self.lbl_J1.setAcceptDrops(False)
self.lbl_J1.setAccessibleName("")
self.lbl_J1.setLayoutDirection(QtCore.Qt.LeftToRight)
self.lbl_J1.setAutoFillBackground(False)
self.lbl_J1.setFrameShape(QtWidgets.QFrame.NoFrame)
self.lbl_J1.setFrameShadow(QtWidgets.QFrame.Plain)
self.lbl_J1.setLineWidth(1)
self.lbl_J1.setScaledContents(False)
self.lbl_J1.setWordWrap(False)
self.lbl_J1.setOpenExternalLinks(False)
self.lbl_J1.setObjectName("lbl_J1")
self.grid_join_full.addWidget(self.lbl_J1, 0, 0, 1, 1)
self.spin_J1 = QtWidgets.QSpinBox(self.centralwidget)
self.spin_J1.setMinimum(-360)
self.spin_J1.setMaximum(360)
self.spin_J1.setProperty("value", 89)

```

```

self.spin_J1.setObjectName("spin_J1")
self.grid_join_full.addWidget(self.spin_J1, 0, 1, 1, 1)
self.btn_des_J6 = QtWidgets.QPushButton(self.centralwidget)
self.btn_des_J6.setStyleSheet("background-color: lightgreen;")
self.btn_des_J6.setObjectName("btn_des_J6")
self.grid_join_full.addWidget(self.btn_des_J6, 5, 3, 1, 1)
self.btn_inc_J6 = QtWidgets.QPushButton(self.centralwidget)
self.btn_inc_J6.setStyleSheet("background-color: lightgreen;")
self.btn_inc_J6.setObjectName("btn_inc_J6")
self.grid_join_full.addWidget(self.btn_inc_J6, 5, 2, 1, 1)
self.spin_J6 = QtWidgets.QSpinBox(self.centralwidget)
self.spin_J6.setMinimum(-360)
self.spin_J6.setMaximum(360)
self.spin_J6.setProperty("value", 89)
self.spin_J6.setObjectName("spin_J6")
self.grid_join_full.addWidget(self.spin_J6, 5, 1, 1, 1)
self.btn_des_J1 = QtWidgets.QPushButton(self.centralwidget)
self.btn_des_J1.setStyleSheet("background-color: lightgreen;")
self.btn_des_J1.setObjectName("btn_des_J1")
self.grid_join_full.addWidget(self.btn_des_J1, 0, 3, 1, 1)
self.lbl_J2 = QtWidgets.QLabel(self.centralwidget)
self.lbl_J2.setObjectName("lbl_J2")
self.grid_join_full.addWidget(self.lbl_J2, 1, 0, 1, 1)
self.spin_J2 = QtWidgets.QSpinBox(self.centralwidget)
self.spin_J2.setMinimum(-360)
self.spin_J2.setMaximum(360)
self.spin_J2.setProperty("value", 89)
self.spin_J2.setObjectName("spin_J2")
self.grid_join_full.addWidget(self.spin_J2, 1, 1, 1, 1)
self.btn_inc_J1 = QtWidgets.QPushButton(self.centralwidget)
self.btn_inc_J1.setStyleSheet("background-color: lightgreen;")
self.btn_inc_J1.setObjectName("btn_inc_J1")
self.grid_join_full.addWidget(self.btn_inc_J1, 0, 2, 1, 1)
self.btn_inc_J2 = QtWidgets.QPushButton(self.centralwidget)
self.btn_inc_J2.setStyleSheet("background-color: lightgreen;")
self.btn_inc_J2.setObjectName("btn_inc_J2")
self.grid_join_full.addWidget(self.btn_inc_J2, 1, 2, 1, 1)
self.btn_des_J2 = QtWidgets.QPushButton(self.centralwidget)

```

```

self.btn_des_J2.setStyleSheet("background-color: lightgreen;")
self.btn_des_J2.setObjectName("btn_des_J2")
self.grid_join_full.addWidget(self.btn_des_J2, 1, 3, 1, 1)
self.rightColumn.addLayout(self.grid_join_full)
self.lbl_htm_big = QtWidgets.QLabel(self.centralwidget)
font = QtGui.QFont()
font.setPointSize(10)
font.setBold(True)
font.setWeight(75)
self.lbl_htm_big.setFont(font)
self.lbl_htm_big.setStyleSheet("font-weight: bold; margin-top:8px;")
self.lbl_htm_big.setAlignment(QtCore.Qt.AlignCenter)
self.lbl_htm_big.setObjectName("lbl_htm_big")
self.rightColumn.addWidget(self.lbl_htm_big)
self.table_htm = QtWidgets.QTableWidget(self.centralwidget)
self.table_htm.setSizeIncrement(QtCore.QSize(4, 4))
self.table_htm.setBaseSize(QtCore.QSize(4, 4))
self.table_htm.setMouseTracking(False)
self.table_htm.setTabletTracking(False)
self.table_htm.setAcceptDrops(True)
self.table_htm.setToolTipDuration(-1)
self.table_htm.setLayoutDirection(QtCore.Qt.LeftToRight)
self.table_htm.setAutoFillBackground(False)
self.table_htm setFrameShape(QtWidgets.QFrame.StyledPanel)
self.table_htm.setLineWidth(0)
self.table_htm.setMidLineWidth(1)
self.table_htm.setVerticalScrollBarPolicy(QtCore.Qt.ScrollBarAsNeeded)
self.table_htm.setHorizontalScrollBarPolicy(QtCore.Qt.ScrollBarAlwaysOff)
self.table_htm.setSizeAdjustPolicy(QtWidgets.QAbstractScrollArea.AdjustToContentsOnFirstShow)
self.table_htm.setAutoScroll(True)
self.table_htm.setAutoScrollMargin(4)
self.table_htm.setTabKeyNavigation(True)
self.table_htm.setProperty("showDropIndicator", True)
self.table_htm.setAlternatingRowColors(False)
self.table_htm.setSelectionMode(QtWidgets.QAbstractItemView.MultiSelection)
self.table_htm.setSelectionBehavior(QtWidgets.QAbstractItemView.SelectColumns)
self.table_htm.setTextElideMode(QtCore.Qt.ElideMiddle)
self.table_htm.setVerticalScrollMode(QtWidgets.QAbstractItemView.ScrollPerPixel)

```

```

self.table_htm.setHorizontalScrollMode(QtWidgets.QAbstractItemView.ScrollPerItem)
self.table_htm.setShowGrid(True)
self.table_htm.setWordWrap(True)
self.table_htm.setRowCount(4)
self.table_htm.setColumnCount(4)
self.table_htm.setObjectName("table_htm")
self.table_htm.horizontalHeader().setVisible(False)
self.table_htm.horizontalHeader().setCascadingSectionResizes(False)
self.table_htm.horizontalHeader().setDefaultSectionSize(198)
self.table_htm.horizontalHeader().setHighlightSections(True)
self.table_htm.horizontalHeader().setMinimumSectionSize(47)
self.table_htm.horizontalHeader().setSortIndicatorShown(False)
self.table_htm.horizontalHeader().setStretchLastSection(True)
self.table_htm.verticalHeader().setVisible(False)
self.table_htm.verticalHeader().setCascadingSectionResizes(False)
self.table_htm.verticalHeader().setDefaultSectionSize(90)
self.table_htm.verticalHeader().setHighlightSections(True)
self.table_htm.verticalHeader().setMinimumSectionSize(24)
self.table_htm.verticalHeader().setSortIndicatorShown(False)
self.table_htm.verticalHeader().setStretchLastSection(True)
self.rightColumn.addWidget(self.table_htm)
self.sliders = QtWidgets.QHBoxLayout()
self.sliders.setObjectName("sliders")
self.lbl_step_dis = QtWidgets.QLabel(self.centralwidget)
self.lbl_step_dis.setObjectName("lbl_step_dis")
self.sliders.addWidget(self.lbl_step_dis)
self.slider_step_dis = QtWidgets.QSlider(self.centralwidget)
self.slider_step_dis.setOrientation(QtCore.Qt.Horizontal)
self.slider_step_dis.setObjectName("slider_step_dis")
self.sliders.addWidget(self.slider_step_dis)
self.lbl_val_step_dis = QtWidgets.QLabel(self.centralwidget)
self.lbl_val_step_dis.setObjectName("lbl_val_step_dis")
self.sliders.addWidget(self.lbl_val_step_dis)
self.lbl_unit_step_dis = QtWidgets.QLabel(self.centralwidget)
self.lbl_unit_step_dis.setObjectName("lbl_unit_step_dis")
self.sliders.addWidget(self.lbl_unit_step_dis)
self.rightColumn.addLayout(self.sliders)
self.sliders2 = QtWidgets.QHBoxLayout()

```

```

self.sliders2.setObjectName("sliders2")
self.lbl_step_rot = QtWidgets.QLabel(self.centralwidget)
self.lbl_step_rot.setObjectName("lbl_step_rot")
self.sliders2.addWidget(self.lbl_step_rot)
self.slider_step_rot = QtWidgets.QSlider(self.centralwidget)
self.slider_step_rot.setOrientation(QtCore.Qt.Horizontal)
self.slider_step_rot.setObjectName("slider_step_rot")
self.sliders2.addWidget(self.slider_step_rot)
self.lbl_val_step_rot = QtWidgets.QLabel(self.centralwidget)
self.lbl_val_step_rot.setObjectName("lbl_val_step_rot")
self.sliders2.addWidget(self.lbl_val_step_rot)
self.lbl_unit_step_rot = QtWidgets.QLabel(self.centralwidget)
self.lbl_unit_step_rot.setObjectName("lbl_unit_step_rot")
self.sliders2.addWidget(self.lbl_unit_step_rot)
self.rightColumn.addLayout(self.sliders2)
self.sliders3 = QtWidgets.QHBoxLayout()
self.sliders3.setObjectName("sliders3")
self.lbl_speed = QtWidgets.QLabel(self.centralwidget)
self.lbl_speed.setObjectName("lbl_speed")
self.sliders3.addWidget(self.lbl_speed)
self.slider_speed = QtWidgets.QSlider(self.centralwidget)
self.slider_speed.setOrientation(QtCore.Qt.Horizontal)
self.slider_speed.setObjectName("slider_speed")
self.sliders3.addWidget(self.slider_speed)
self.lbl_val_speed = QtWidgets.QLabel(self.centralwidget)
self.lbl_val_speed.setObjectName("lbl_val_speed")
self.sliders3.addWidget(self.lbl_val_speed)
self.btn_speed_set = QtWidgets.QPushButton(self.centralwidget)
self.btn_speed_set.setObjectName("btn_speed_set")
self.sliders3.addWidget(self.btn_speed_set)
self.rightColumn.addLayout(self.sliders3)
self.mainContent.addLayout(self.rightColumn)
self.verticalLayout.addLayout(self.mainContent)
MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QtWidgets.QMenuBar(MainWindow)
self.menubar.setGeometry(QtCore.QRect(0, 0, 1102, 26))
self.menubar.setObjectName("menubar")
MainWindow.setMenuBar(self.menubar)

```

```
self.statusbar = QtWidgets.QStatusBar(MainWindow)
self.statusbar.setObjectName("statusbar")
MainWindow.setStatusBar(self.statusbar)
self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)
def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "Robot 6DOF Control"))
    self.btn_close.setText(_translate("MainWindow", "X"))
    self.titleLabel.setText(_translate("MainWindow", "Robot 6DOF Control"))
    self.lbl_position.setText(_translate("MainWindow", "POSITION CONTROL"))
    self.btn_z_plus.setText(_translate("MainWindow", "↑ Z+"))
    self.btn_o.setText(_translate("MainWindow", "O"))
    self.btn_z_minus.setText(_translate("MainWindow", "↓ Z-"))
    self.btn_x_plus.setText(_translate("MainWindow", "✓ X+"))
    self.btn_x_minus.setText(_translate("MainWindow", "X-"))
    self.lbl_tcp.setText(_translate("MainWindow", "TCP CONTROL"))
    self.btn_home_small.setText(_translate("MainWindow", "HOME"))
    self.btn_connect.setText(_translate("MainWindow", "Connect COM"))
    self.lbl_join_big.setText(_translate("MainWindow", "JOIN CONTROL"))
    self.btn_inc_J3.setText(_translate("MainWindow", "INC ↑"))
    self.btn_des_J3.setText(_translate("MainWindow", "DES ↓"))
    self.lbl_J3.setText(_translate("MainWindow", "JOIN 3"))
    self.btn_des_J4.setText(_translate("MainWindow", "DES ↓"))
    self.btn_inc_J4.setText(_translate("MainWindow", "INC ↑"))
    self.lbl_J4.setText(_translate("MainWindow", "JOIN 4"))
    self.lbl_J5.setText(_translate("MainWindow", "JOIN 5"))
    self.btn_des_J5.setText(_translate("MainWindow", "DES ↓"))
    self.btn_inc_J5.setText(_translate("MainWindow", "INC ↑"))
    self.lbl_J6.setText(_translate("MainWindow", "JOIN 6"))
    self.lbl_J1.setText(_translate("MainWindow", "JOIN 1"))
    self.btn_des_J6.setText(_translate("MainWindow", "DES ↓"))
    self.btn_inc_J6.setText(_translate("MainWindow", "INC ↑"))
    self.btn_des_J1.setText(_translate("MainWindow", "DES ↓"))
    self.lbl_J2.setText(_translate("MainWindow", "JOIN 2"))
    self.btn_inc_J1.setText(_translate("MainWindow", "INC ↑"))
    self.btn_inc_J2.setText(_translate("MainWindow", "INC ↑"))
    self.btn_des_J2.setText(_translate("MainWindow", "DES ↓"))
```

```

self.lbl_htm_big.setText(_translate("MainWindow", "HOMOGENEOUS TRANSFORMATION MATRICES"))
self.table_htm.setSortingEnabled(False)
self.lbl_step_dis.setText(_translate("MainWindow", "STEP-DIS"))
self.lbl_val_step_dis.setText(_translate("MainWindow", "1"))
self.lbl_unit_step_dis.setText(_translate("MainWindow", "CM"))
self.lbl_step_rot.setText(_translate("MainWindow", "STEP-ROT"))
self.lbl_val_step_rot.setText(_translate("MainWindow", "2"))
self.lbl_unit_step_rot.setText(_translate("MainWindow", "DEG"))
self.lbl_speed.setText(_translate("MainWindow", "SPEED LEVEL"))
self.lbl_val_speed.setText(_translate("MainWindow", "1"))
self.btn_speed_set.setText(_translate("MainWindow", "SET"))

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())

```

Chương trình chính

```

# codedieukhien.py
import sys
import serial
import serial.tools.list_ports
import numpy as np
from PyQt5 import QtWidgets, QtCore
from robot_control import Ui_MainWindow
# ===== Cấu hình mặc định (sửa nếu cần) =====
DEFAULT_SERIAL_PORT = 'COM3'
BAUD_RATE = 115200
# Bảng bước xung bạn cung cấp (bước cho 10°) -> chia cho 10 => bước / 1°
PULSE_STEP_PER_10DEG = [72.222, 50.0, 116.7, 105.55, 111.11, 105.55]
PULSE_STEP_PER_DEG = [v / 10.0 for v in PULSE_STEP_PER_10DEG]
# Mặc định: xung "zero" (ở đây chọn 1500us làm trung tâm). Nếu servo thực tế khác, chỉnh list sau.
PULSE_ZERO_OFFSET = [1500, 1500, 1500, 1500, 1500, 1500]
# Kích thước khung (theo yêu cầu): L1 = 17, L2 = 13 (đã chỉnh), L3 = 3 (đơn vị cm)
# Khi tính HTM mình sẽ dùng mét (m)

```

```

L1_cm = 17.0
L2_cm = 13.0 # <-- đã chỉnh theo yêu cầu
L3_cm = 0
# Chuyển sang mét
L1 = L1_cm * 10
L2 = L2_cm * 10
L3 = L3_cm * 10

# Hàm chuyển góc -> xung cho từng servo
def angle_to_pulse(servo_index: int, angle_deg: float) -> int:
    """Map góc (0..180) sang pulse theo hệ số riêng của từng servo.
    pulse = zero_offset + (angle - 90) * step_per_deg
    (trong đó mình dựa vào trung tâm 90° là offset 1500)"""
    angle = max(0.0, min(180.0, float(angle_deg)))
    step = PULSE_STEP_PER_DEG[servo_index]
    zero = PULSE_ZERO_OFFSET[servo_index]
    # Tính tương đối so với 90°
    pulse = zero + (angle - 90.0) * step
    return int(round(pulse))

class RobotController(QtWidgets.QMainWindow):
    def __init__(self):
        super().__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        # Serial
        self.ser = None
        self.current_port = None
        self.baud = BAUD_RATE
        # SpinBox góc (tên chính xác từ robot_control.py)
        self.joint_spinboxes = [
            self.ui.spin_J1,
            self.ui.spin_J2,
            self.ui.spin_J3,
            self.ui.spin_J4,
            self.ui.spin_J5,
            self.ui.spin_J6
        ]
        # Nút tăng (btn_inc_Jn)

```



```

self.buttons_inc = [
    self.ui.btn_inc_J1,
    self.ui.btn_inc_J2,
    self.ui.btn_inc_J3,
    self.ui.btn_inc_J4,
    self.ui.btn_inc_J5,
    self.ui.btn_inc_J6
]

# Nút giảm (btn_des_Jn)
self.buttons_des = [
    self.ui.btn_des_J1,
    self.ui.btn_des_J2,
    self.ui.btn_des_J3,
    self.ui.btn_des_J4,
    self.ui.btn_des_J5,
    self.ui.btn_des_J6
]

# Gán sự kiện tăng/giảm: dùng step từ slider_step_rot (deg)
for i in range(6):
    self.buttons_inc[i].clicked.connect(lambda checked, j=i: self.adjust_joint(j, +self.get_step_rot()))
    self.buttons_des[i].clicked.connect(lambda checked, j=i: self.adjust_joint(j, -self.get_step_rot()))

# Connect / Disconnect button
self.ui.btn_connect.clicked.connect(self.toggle_connect)

# Home button
self.ui.btn_home_small.clicked.connect(self.move_home)

# If there's a "btn_setting" in UI, connect it to send_all_joints
if hasattr(self.ui, "btn_setting"):
    self.ui.btn_setting.clicked.connect(self.send_all_joints)

# Speed slider initial
if hasattr(self.ui, "slider_speed"):
    self.ui.slider_speed.setMinimum(100)
    self.ui.slider_speed.setMaximum(5000)
    self.ui.slider_speed.setValue(1000)
    self.ui.slider_speed.valueChanged.connect(self.on_speed_changed)
    self.current_speed = int(self.ui.slider_speed.value())
else:
    self.current_speed = 1000

# slider_step_rot show value if label present

```

```

if hasattr(self.ui, "slider_step_rot") and hasattr(self.ui, "lbl_val_step_rot"):
    self.ui.slider_step_rot.setMinimum(1)
    self.ui.slider_step_rot.setMaximum(30)
    self.ui.slider_step_rot.setValue(2)
    self.ui.slider_step_rot.valueChanged.connect(lambda v: self.ui.lbl_val_step_rot.setText(str(v)))
# Khi spinbox thay đổi => cập nhật HTM
for sb in self.joint_spinboxes:
    sb.valueChanged.connect(self.update_hm_table)
# Khởi tạo table_hm nếu có
if hasattr(self.ui, "table_hm"):
    self.ui.table_hm.setRowCount(4)
    self.ui.table_hm.setColumnCount(4)
    for i in range(4):
        for j in range(4):
            self.ui.table_hm.setItem(i, j, QtWidgets.QTableWidgetItem("0.000000"))
# Cập nhật label nút connect ban đầu
self.update_connect_button_label()
# Update HTM lần đầu
self.update_hm_table()
# ----- Serial helpers -----
def update_connect_button_label(self):
    if self.ser and self.ser.is_open:
        self.ui.btn_connect.setText(f"Disconnect ( {self.current_port} )")
    else:
        ports = list(serial.tools.list_ports.comports())
        if ports:
            # show up to 3 port names
            names = ", ".join([p.device for p in ports][:3])
            self.ui.btn_connect.setText(f"Connect ( {names} )")
        else:
            self.ui.btn_connect.setText("Connect (no ports)")

def toggle_connect(self):
    # If connected -> disconnect
    if self.ser and self.ser.is_open:
        try:
            self.ser.close()
        except Exception:

```

```

        pass
        self.ser = None
        self.current_port = None
        print("🔌 Disconnected.")
        self.update_connect_button_label()
        return

# try to find ports
ports = list(serial.tools.list_ports.comports())
if not ports:
    QtWidgets.QMessageBox.warning(self, "No COM", "Không tìm thấy cổng COM. Cắm thiết bị vào rồi thử lại.")
    self.update_connect_button_label()
    return

# try default port first, else first found
port_to_try = DEFAULT_SERIAL_PORT
found = False
for p in ports:
    if p.device.upper() == DEFAULT_SERIAL_PORT.upper():
        port_to_try = p.device
        found = True
        break
if not found:
    port_to_try = ports[0].device
try:
    self.ser = serial.Serial(port_to_try, self.baud, timeout=1)
    self.current_port = port_to_try
    print(f"✅ Connected to {port_to_try} @ {self.baud}")
except Exception as e:
    QtWidgets.QMessageBox.critical(self, "Serial error", f"Không thể mở {port_to_try}: {e}")
    self.ser = None
    self.current_port = None
    self.update_connect_button_label()
def on_speed_changed(self, v):
    self.current_speed = int(v)
    if hasattr(self.ui, "lbl_val_speed"):
        self.ui.lbl_val_speed.setText(str(self.current_speed))
    print(f"Speed = {self.current_speed}")
# ----- Servo command -----
def send_servo(self, servo_index: int, angle_deg: float, speed: int = None):

```

```

"""Send servo command formatted as #<id>P<pulse>T<speed>\r\n"""
if speed is None:
    speed = self.current_speed
if not (0 <= servo_index <= 6):
    print("Invalid servo index", servo_index)
    return
pulse = angle_to_pulse(servo_index, angle_deg)
cmd = f'#{servo_index+1}P{pulse}T{int(speed)}\r\n'
if self.ser and self.ser.is_open:
    try:
        self.ser.write(cmd.encode("ascii"))
        print("Gửi:", cmd.strip())
    except Exception as e:
        print("Lỗi khi gửi serial:", e)
else:
    # Khi chưa nối, in ra để debug
    print("Serial chưa mở — (simulate) Gửi:", cmd.strip())
# ----- Joint control -----
def get_step_rot(self):
    if hasattr(self.ui, "slider_step_rot"):
        return int(self.ui.slider_step_rot.value())
    return 2
def adjust_joint(self, joint_index: int, delta_deg: float):
    sb = self.joint_spinboxes[joint_index]
    new_val = sb.value() + delta_deg
    new_val = max(0, min(180, new_val)) # giới hạn 0..180
    sb.setValue(int(new_val))
    self.send_servo(joint_index, new_val, self.current_speed)
    self.update_htm_table()
def send_all_joints(self):
    for i, sb in enumerate(self.joint_spinboxes):
        self.send_servo(i, sb.value(), self.current_speed)

def move_home(self):
    for i, sb in enumerate(self.joint_spinboxes):
        sb.setValue(90)
        self.send_servo(i, 90, self.current_speed)
    self.update_htm_table()

```

```
# ----- Kinematics: DH and HTM -----

def dh_transform(self, a: float, alpha: float, d: float, theta: float):
    ct = np.cos(theta)
    st = np.sin(theta)
    ca = np.cos(alpha)
    sa = np.sin(alpha)
    return np.array([
        [ct, -st * ca, st * sa, a * ct],
        [st, ct * ca, -ct * sa, a * st],
        [0, sa, ca, d],
        [0, 0, 0, 1]
    ])

def update_htm_table(self):
    # Lấy góc khớp (deg) -> rad
    theta_deg = [int(sb.value()) for sb in self.joint_spinboxes]
    theta_rad = [np.radians(t) for t in theta_deg]

    # Bảng DH theo L1,L2,L3 (đã chuyển sang mét)
    # Mình dùng ký hiệu: a_i, alpha_i, d_i, theta_i
    # NOTE: thiết lập DH tùy robot — dưới đây là theo doc bạn gửi (3 khớp đầu)
    DH = [
        # a,    alpha,    d,    theta
        [0.0,  np.pi/2,  L1,    theta_rad[0]], # joint1
        [L2,   0.0,     0.0,    theta_rad[1]], # joint2
        [L3,   0.0,     0.0,    theta_rad[2]], # joint3
        # cho khớp 4..6 mình giữ cấu trúc dạng giả định (xoay/wrist)
        [0.0,  np.pi/2,  0.0,    theta_rad[3]],
        [0.0,  -np.pi/2, 0.0,    theta_rad[4]],
        [0.0,  0.0,     0.0,    theta_rad[5]]
    ]

    # Tích liên tiếp
    T = np.eye(4)
    for (a, alpha, d, theta) in DH:
        T = T @ self.dh_transform(a, alpha, d, theta)

    # Hiện thị lên bảng table_htm (4x4) nếu có
    if hasattr(self.ui, "table_htm"):
        for i in range(4):
            for j in range(4):
```

```
        val = float(T[i, j])
        self.ui.table_htm.setItem(i, j, QtWidgets.QTableWidgetItem(f'{val:.6f}'))

    # (Tùy chọn) in ra để kiểm tra
    print("HTM (end-effector):\n", np.round(T, 6))

# ----- Main -----
if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    win = RobotController()
    win.show()
    sys.exit(app.exec_())
```