



BÁO CÁO ROBOT SCARA

NGUYỄN PHONG THÁI - 20226429

KỸ THUẬT LẬP TRÌNH TRONG CƠ ĐIỆN TỬ

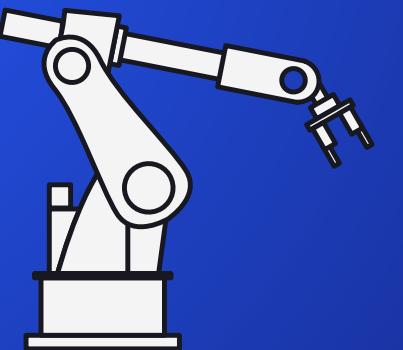
Mã lớp: 157018

Mã HP: ME3300

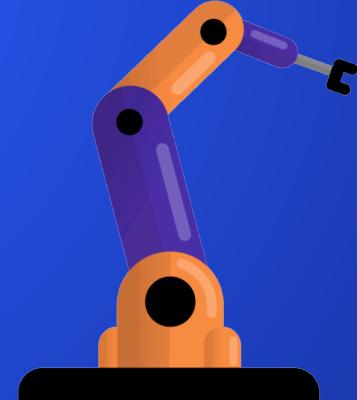
GVHD: TS. NGUYỄN XUÂN THUẬN



NỘI DUNG BÁO CÁO



PHẦN CỨNG



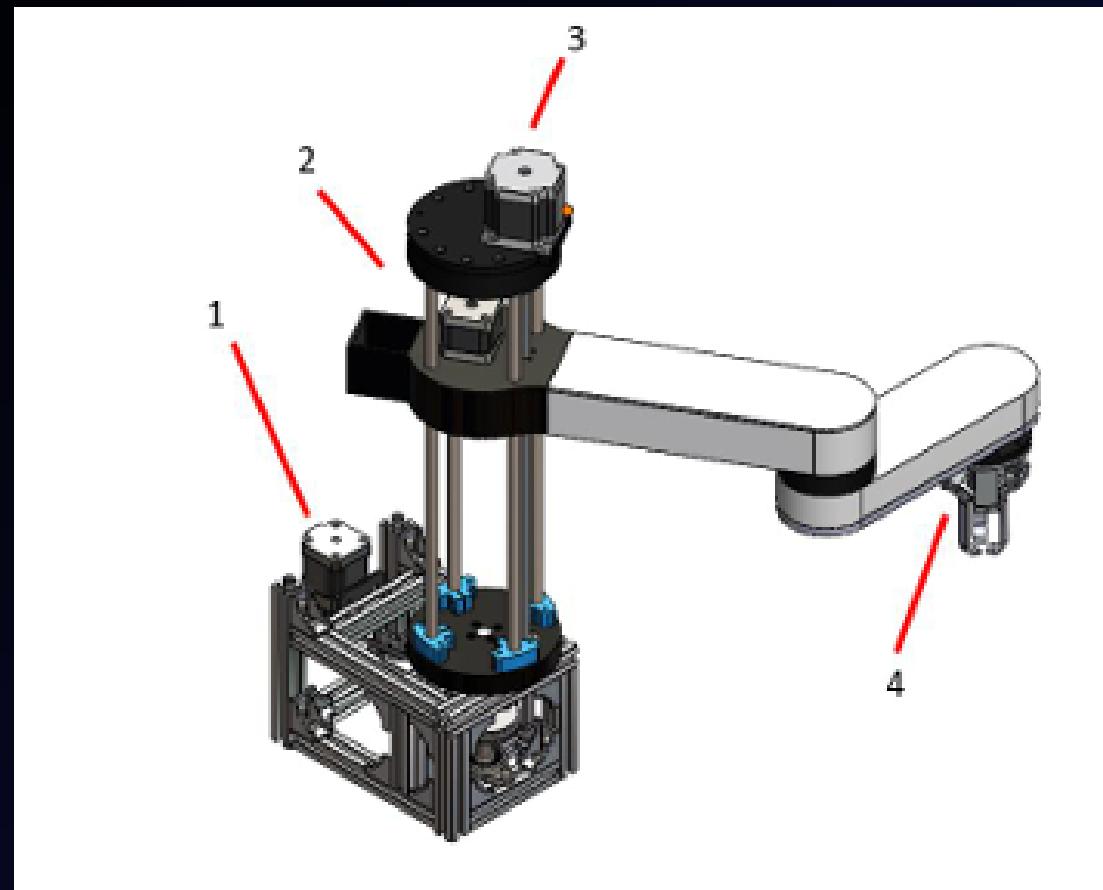
ĐIỀU KHIỂN - CODE



GIAO DIỆN GUI

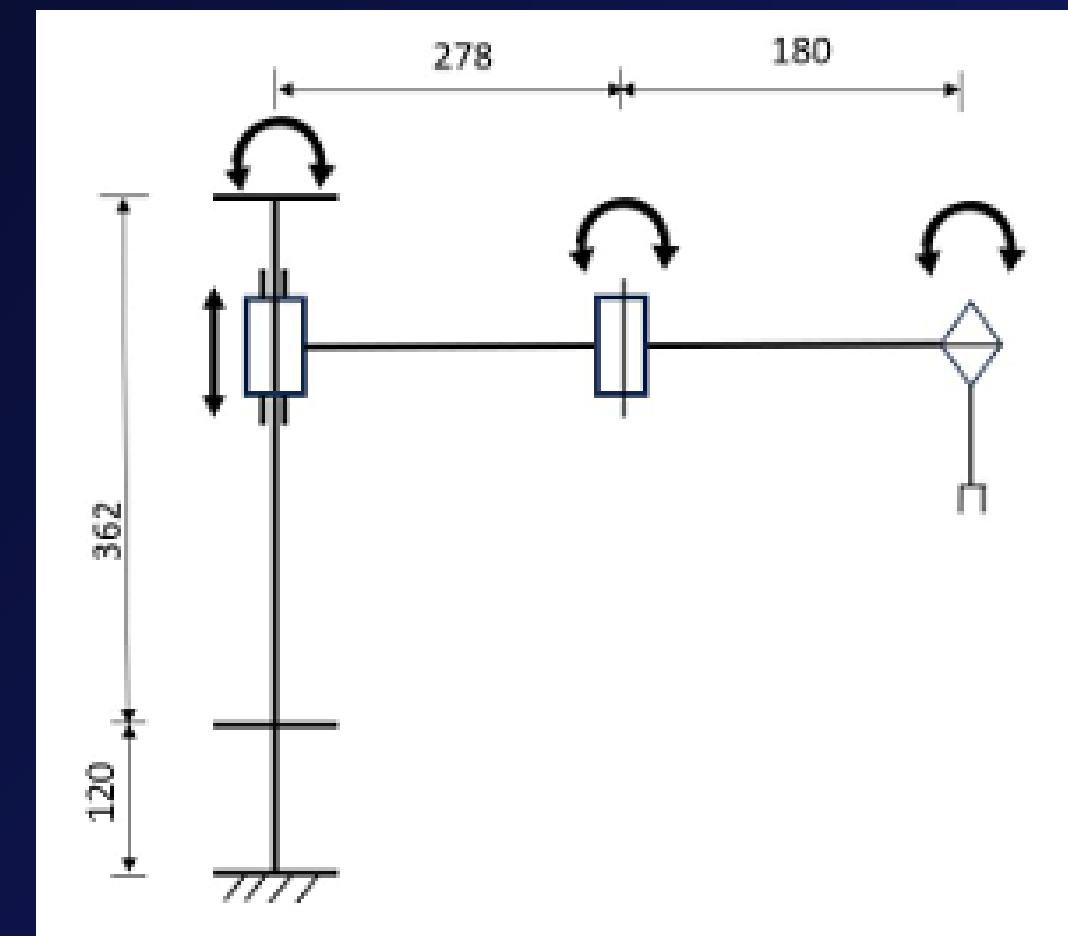
ROBOT SCARA

PHÂN TÍCH ROBOT



Mô hình 3D

Robot SCARA có cấu trúc gồm 4 bộ phận chính: đế robot, hai cánh tay nối (khâu 1 và khâu 2) và cơ cầu chìa hành ở đầu cuối. Các khâu có khả năng quay quanh trục Z, cho phép robot di chuyển linh hoạt trong mặt phẳng XY và thực hiện thao tác chính xác.



Kích thước các khâu

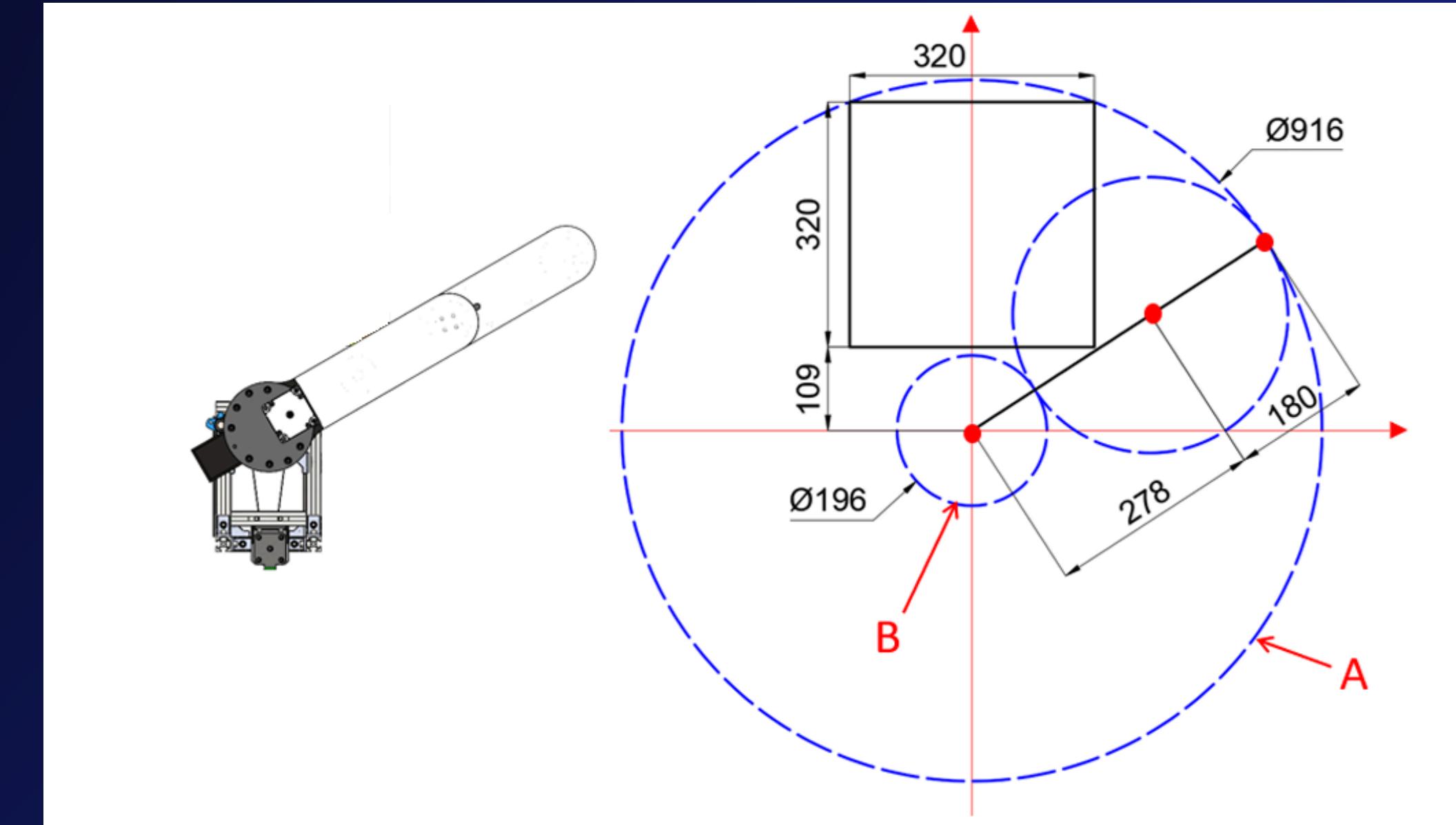
Chiều dài khâu 1 là 278 mm, khâu 2 là 180 mm, chiều cao tổng thể khoảng 382 mm. Các khâu được thiết kế xoay để phù hợp với nhiệm vụ lắp ráp nhanh và chính xác.

ROBOT SCARA

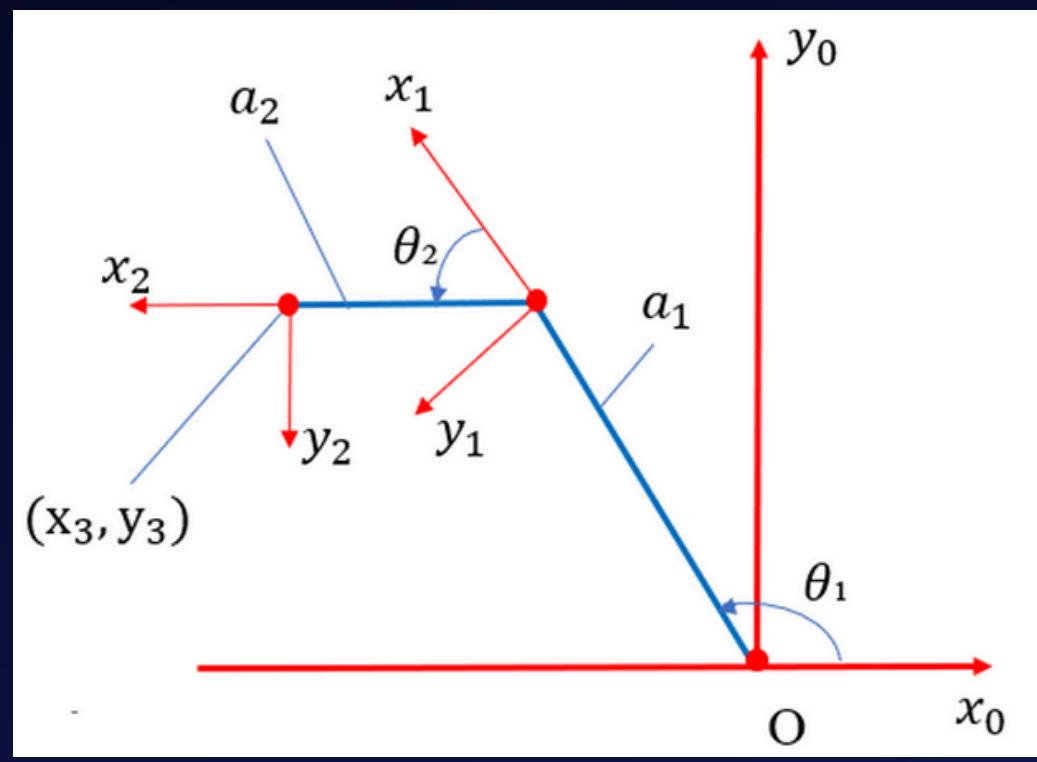
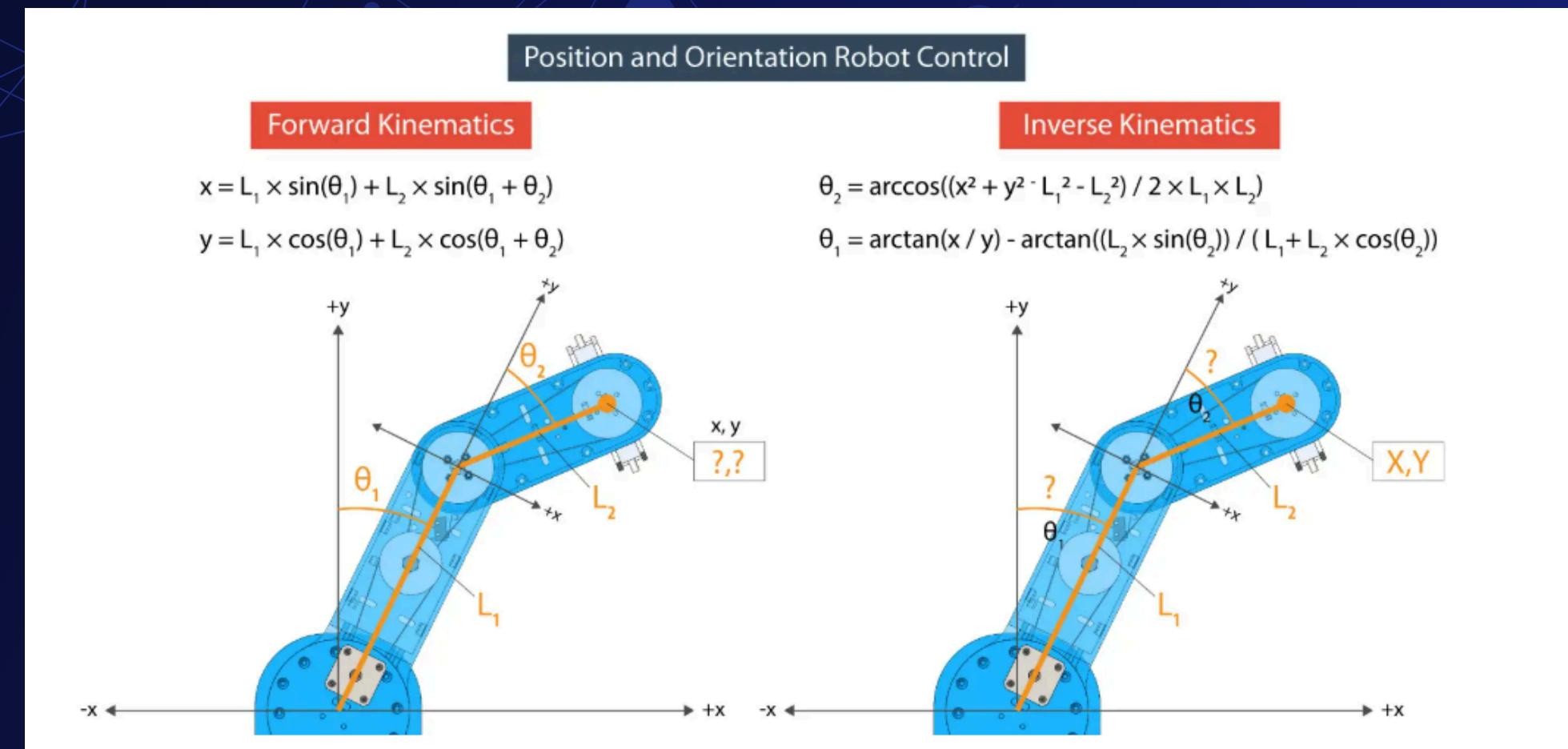
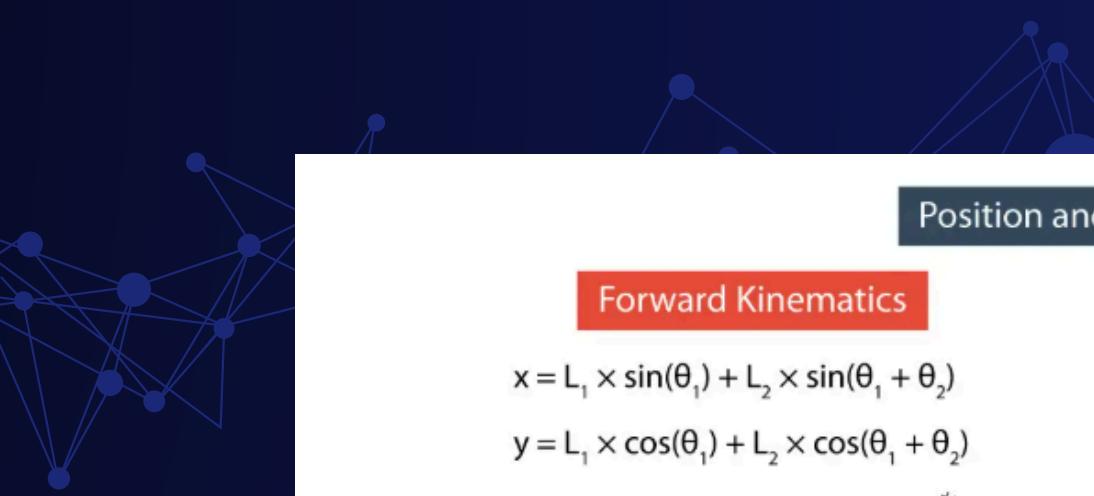
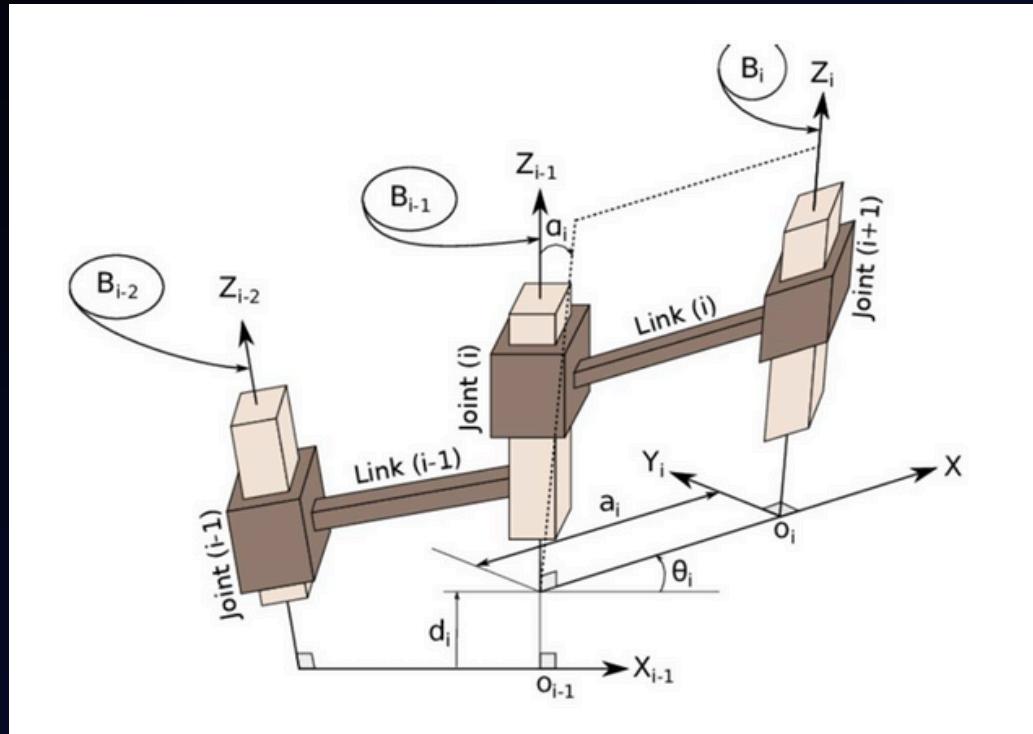
PHÂN TÍCH ROBOT

Sơ đồ quỹ đạo chuyển động

Hình mô tả kích thước của khâu 1, khâu 2 trên mặt phẳng Oxy (đường màu đen). Quỹ đạo chuyển động là các đường màu xanh. Tầm với xa nhất của robot là đường tròn có đường kính 916 mm (A) từ tâm và tầm với nhỏ nhất là đường tròn đường kính 196 mm (B).



ROBOT SCARA TÍNH TOÁN ĐỘNG HỌC



Công thức tính động học thuận - động học nghịch

Bảng DH

Khâu	θ	d	a	α
1	θ_1	0	a_1	0
2	θ_2	0	a_2	0

ROBOT SCARA PHẦN CỨNG



Step Motor NEMA 17 & NEMA 23

NEMA 17: Moment xoắn giữ vị trí từ 0.3 – 0.7 kg.cm.
NEMA 23: Kích thước 57x57x57 mm, moment xoắn giữ vị trí từ 0.5 – 3 Nm.



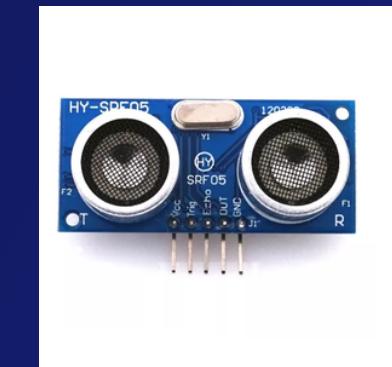
Bộ truyền Puly - Đai răng

Tỷ số truyền 57/11 * 80/20
(Dùng cho stepmotor điều khiển góc xoay khâu 1 và 2)



Bộ truyền vít me - đai ốc

Vít me T8, bước ren 6mm
(Dùng cho trục Z)



Công tắc hành trình

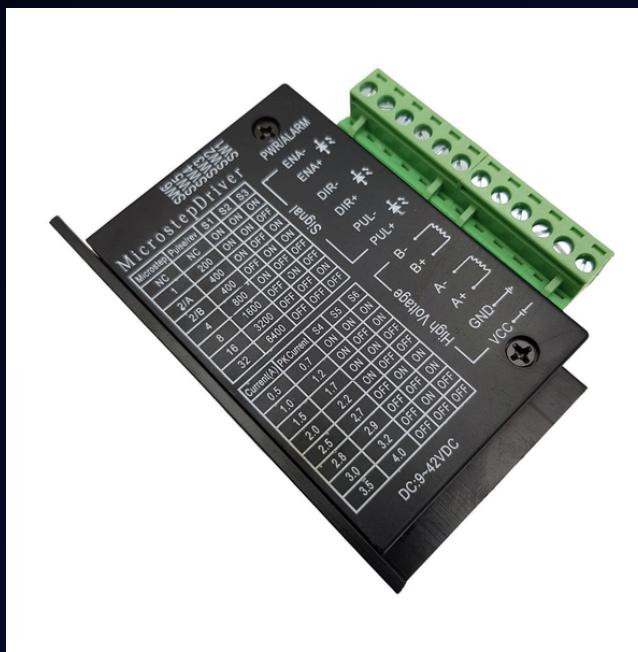


Cảm biến siêu âm HCSR05

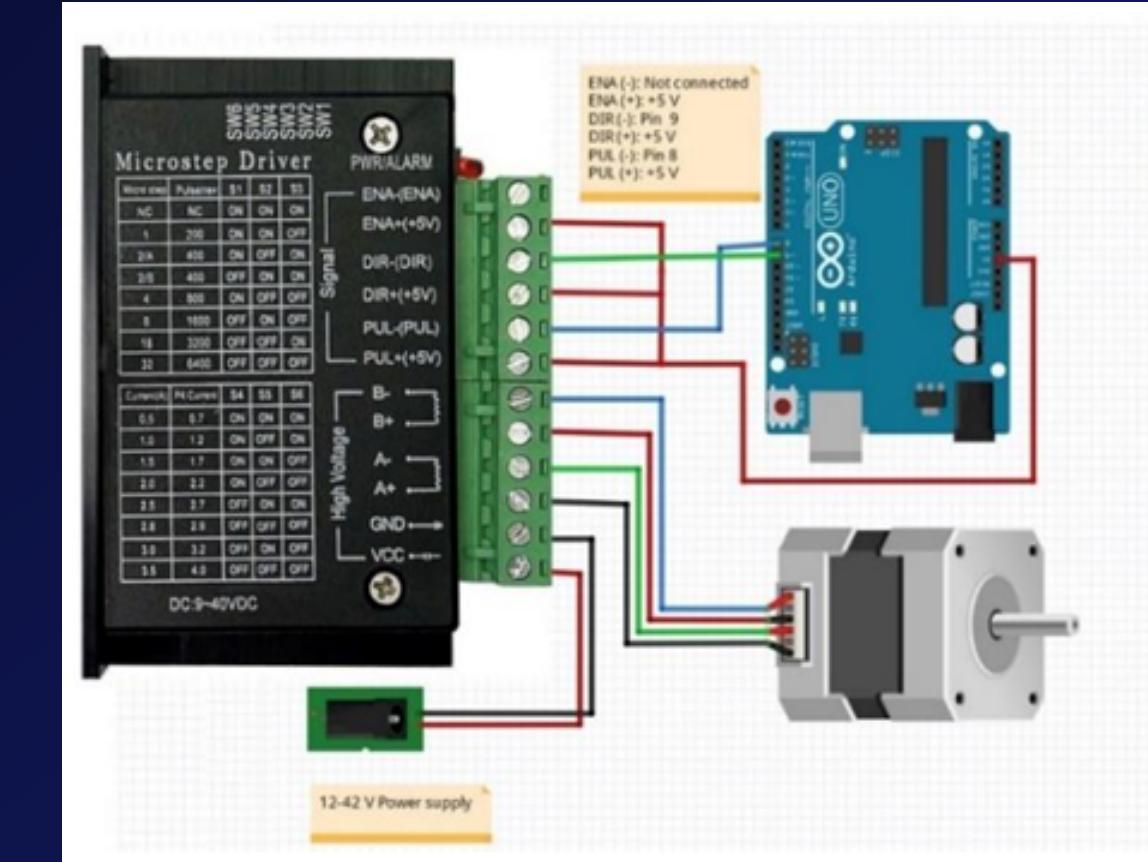


ROBOT SCARA

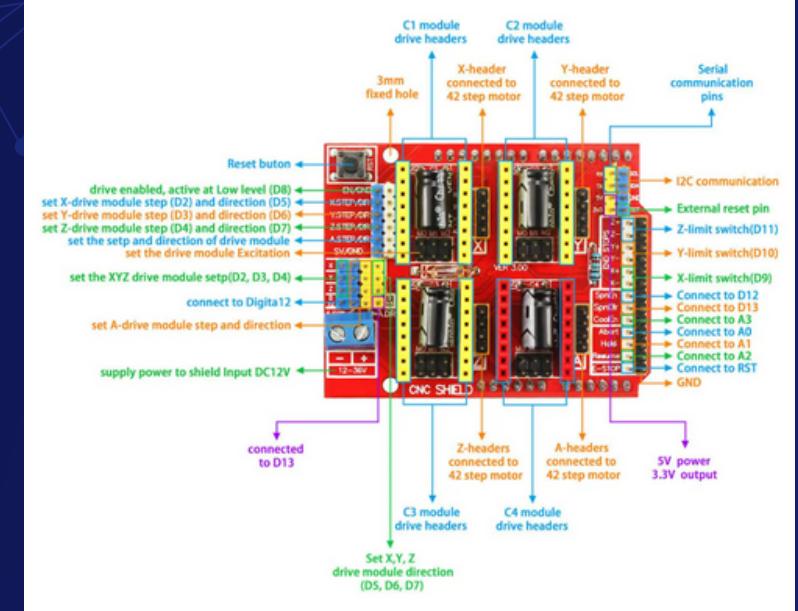
PHẦN CỨNG



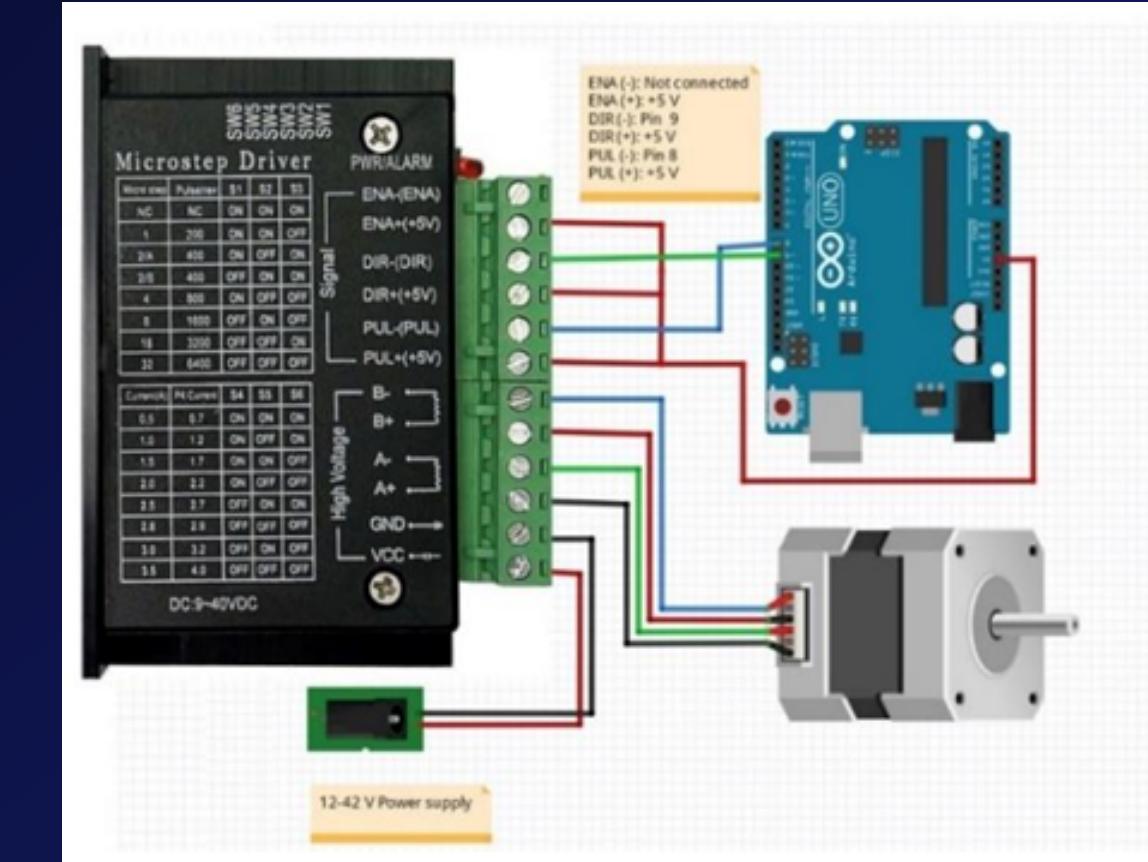
Driver TB6600



Vi điều khiển Arduino



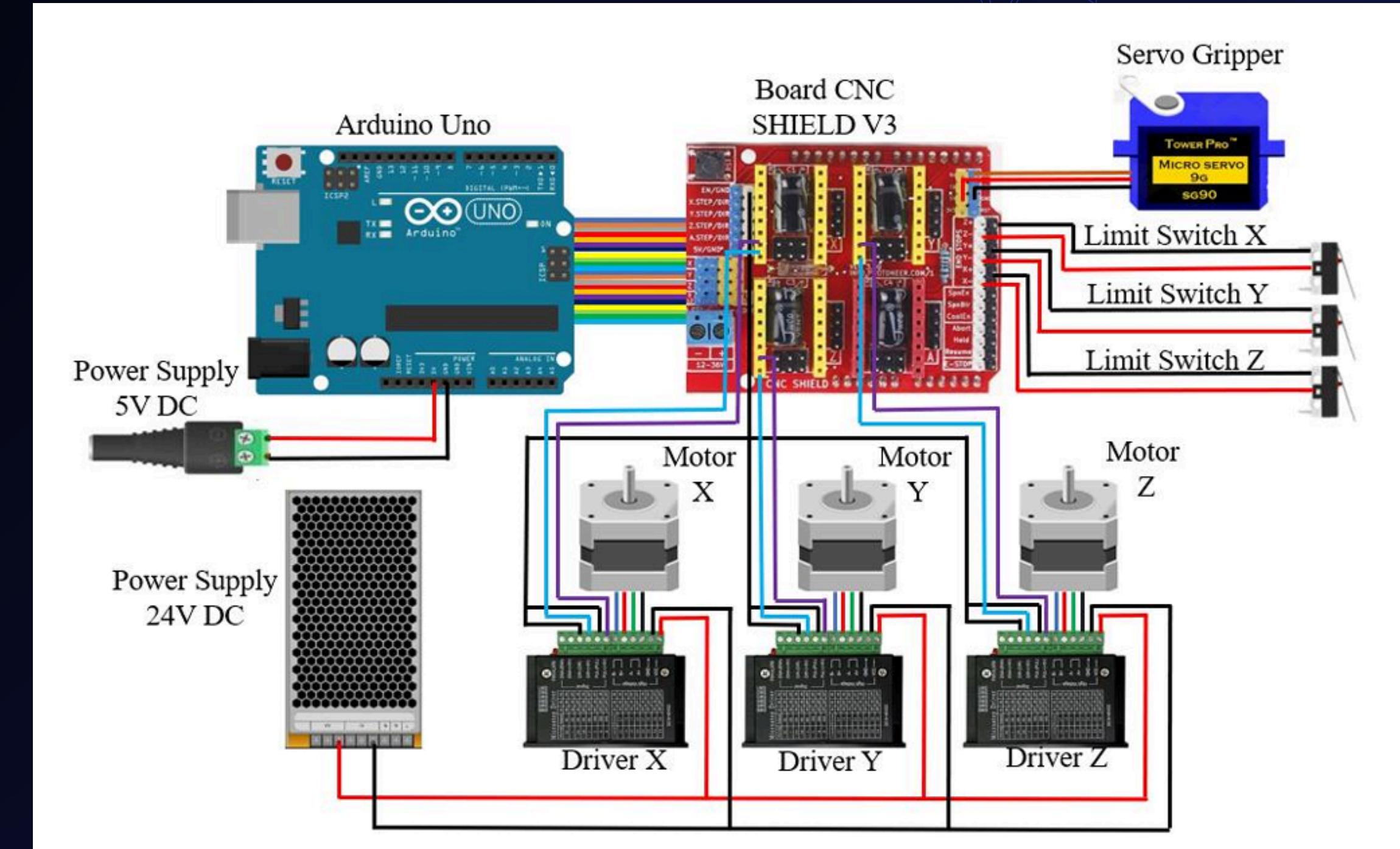
Mạch CNC SHILED V3



Sơ đồ chạy thử

ROBOT SCARA

ĐIỀU KHIỂN - CODE



SƠ ĐỒ ĐẦU NỐI TỔNG THỂ

ROBOT SCARA

ĐIỀU KHIỂN - CODE

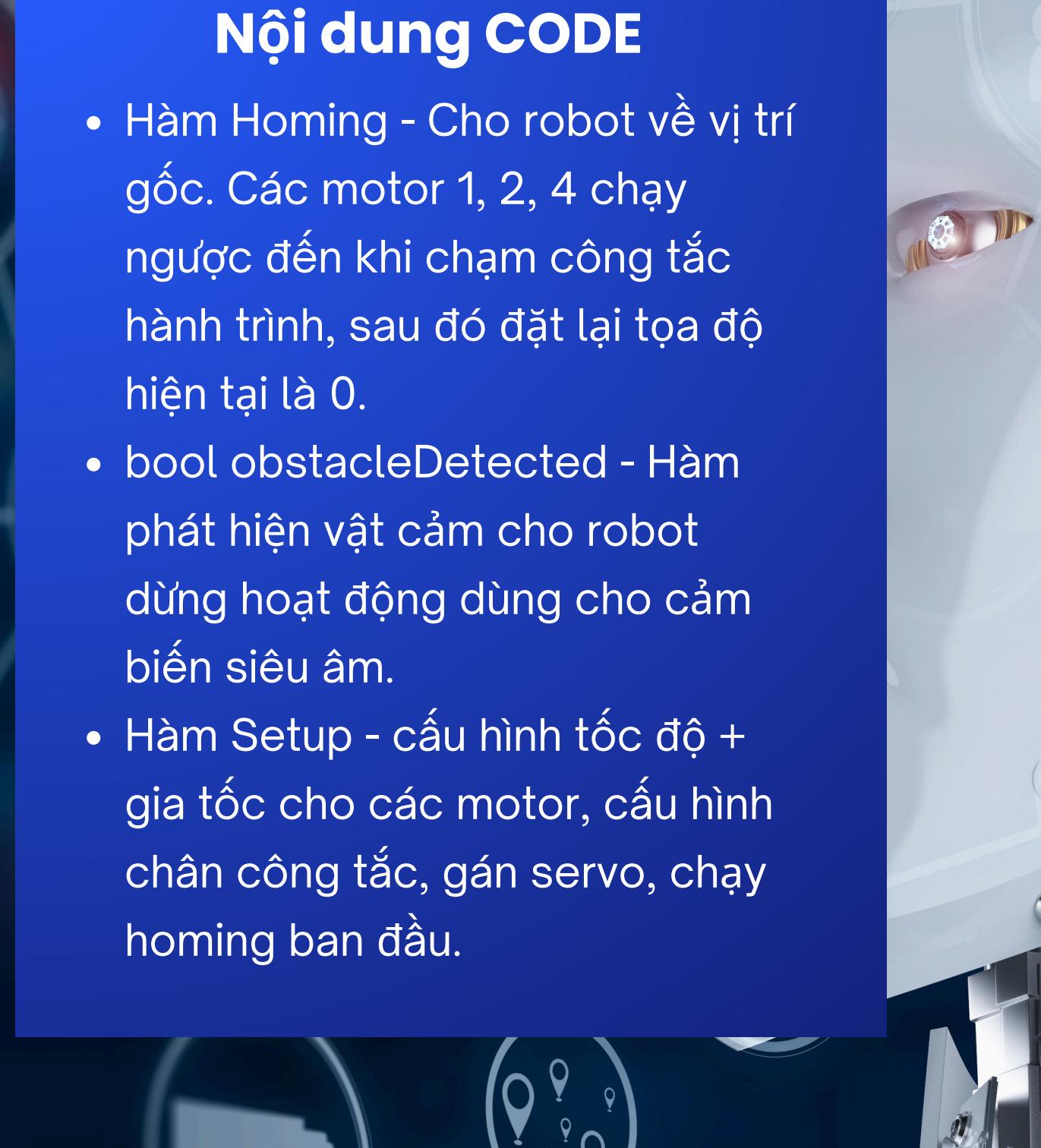
Nội dung CODE

- ➔ • Thư viện AccelStepper - điều khiển tốc độ và gia tốc stepmotor, Servo - điều khiển servo tay gấp, NewPing - cảm biến siêu âm.
- Cài đặt chân điều khiển step motor, công tắc hành trình và cảm biến siêu âm.
- Tỉ số truyền của từng bộ truyền (Dùng để tính toán động học).

```
SCARA_Robot2.ino
1 #include <AccelStepper.h>
2 #include <Servo.h>
3 #include <NewPing.h>
4 // Stepper drivers
5 AccelStepper stepper1(AccelStepper::DRIVER, 2, 5);
6 AccelStepper stepper2(AccelStepper::DRIVER, 3, 6);
7 AccelStepper stepper3(AccelStepper::DRIVER, 12,13);
8 AccelStepper stepper4(AccelStepper::DRIVER, 4, 7);
9 Servo gripperServo;
10 // Limit switches
11 #define LIMIT_SWITCH_PIN_1 9
12 #define LIMIT_SWITCH_PIN_2 10
13 #define LIMIT_SWITCH_PIN_3 11
14 // HC-SR04 ultrasonic sensor
15 #define SRF05_TRIG_PIN A1
16 #define SRF05_ECHO_PIN A2
17 #define SRF05_MAX_CM 50
18 NewPing sonar(SRF05_TRIG_PIN, SRF05_ECHO_PIN, SRF05_MAX_CM);
19 // Kinematics constants
20 const float TH1 = 92.12f;
21 const float TH2 = 71.111111f;
22 const float PHI = 0.0f;
23 const float ZSTEP = 200.0f;
24 // Command parsing and state machine
25 enum State { IDLE, MANUAL_STEP, TRAJECTORY } state = IDLE;
26 enum TrajPhase { PHASE1, PHASE2 } trajPhase;
27 int bufData[10];
28 int manualID, manualStep;
29 int trajMode;
30 long tarPos[4];
31 int gripVal, speedVal, accelVal;
32 // Obstacle detection timing
33 unsigned long lastPing = 0;
34 int lastCm = -1;
35 // Homing routine
void manualHoming() {
```

ROBOT SCARA

ĐIỀU KHIỂN - CODE



Nội dung CODE

- Hàm Homing - Cho robot về vị trí gốc. Các motor 1, 2, 4 chạy ngược đến khi chạm công tắc hành trình, sau đó đặt lại tọa độ hiện tại là 0.
- bool obstacleDetected - Hàm phát hiện vật cản cho robot dừng hoạt động dùng cho cảm biến siêu âm.
- Hàm Setup - cấu hình tốc độ + gia tốc cho các motor, cấu hình chân công tắc, gán servo, chạy homing ban đầu.

```
A_Robot2.ino

// Homing routine
void manualHoming() {
    stepper1.setSpeed(-1000);
    stepper2.setSpeed(-1000);
    stepper4.setSpeed(-2000);
    bool d1=false, d2=false, d4=false;
    while (!(d1&&d2&&d4)) {
        if (!d1) { if (digitalRead(LIMIT_SWITCH_PIN_1)==HIGH) stepper1.runSpeed(); else { stepper1.setCurrentPosition(0); d1=1; } }
        if (!d2) { if (digitalRead(LIMIT_SWITCH_PIN_2)==HIGH) stepper2.runSpeed(); else { stepper2.setCurrentPosition(0); d2=1; } }
        if (!d4) { if (digitalRead(LIMIT_SWITCH_PIN_3)==HIGH) stepper4.runSpeed(); else { stepper4.setCurrentPosition(0); d4=1; } }
    }
    gripperServo.write(0);
    Serial.println("Homing complete");
}

bool obstacleDetected() {
    unsigned long now = millis();
    if (now - lastPing >= 20) {
        lastPing = now;
        lastCm   = sonar.ping_cm();
    }
    return (lastCm > 0 && lastCm <= 3);
}

void setup() {
    Serial.begin(115200);

    // Configure steppers
    stepper1.setMaxSpeed(4000); stepper1.setAcceleration(2000);
    stepper2.setMaxSpeed(4000); stepper2.setAcceleration(2000);
    stepper3.setMaxSpeed(4000); stepper3.setAcceleration(2000);
    stepper4.setMaxSpeed(4000); stepper4.setAcceleration(2000);

    // Limit switches
    pinMode(LIMIT_SWITCH_PIN_1, INPUT_PULLUP);
    pinMode(LIMIT_SWITCH_PIN_2, INPUT_PULLUP);
    pinMode(LIMIT_SWITCH_PIN_3, INPUT_PULLUP);

    // Servo
    gripperServo.attach(A0, 600, 2500);
    gripperServo.write(0);

    // Initial homing
    manualHoming();
}
```

ROBOT SCARA

ĐIỀU KHIỂN - CODE

Nội dung CODE

→ Hàm Loop(): điều khiển theo trạng thái

- Kiểm tra vật cản:

Nếu có vật cản gần → dừng robot + chạy lại homing
+ chuyển về trạng thái IDLE.

- MANUAL_STEP:

Robot chạy 1 bước nhỏ cho 1 trục theo lệnh thủ công.

- TRAJECTORY:

Robot thực hiện quỹ đạo chuyển động tự động với 3 chế độ:

trajMode == 1: chạy đồng thời XY và Z

trajMode == 2: chạy Z trước, sau đó chạy XY

trajMode == 3: chạy XY trước, sau đó chạy Z

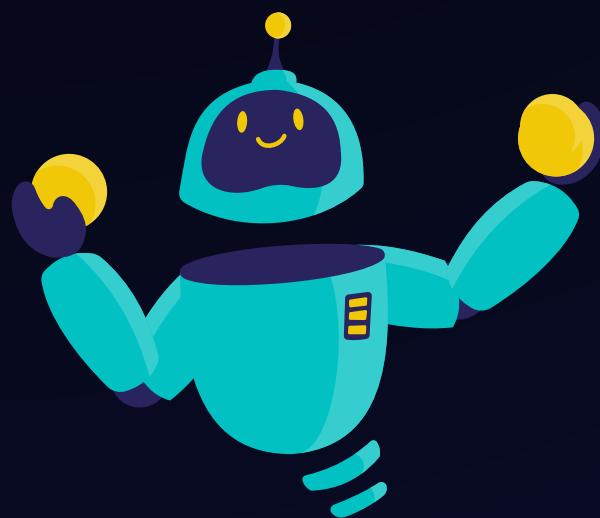
Sau khi xong thì kích hoạt gripper (servo) và trở lại trạng thái IDLE.

IDLE – đọc lệnh từ Serial:

```
SCARA_Robot2.ino
77 }
78 void loop() {
79     // 1) Global obstacle check
80     if (obstacleDetected()) {
81         Serial.println("OBJECT_TOO_CLOSE");
82         // stop all
83         stepper1.stop(); stepper2.stop(); stepper3.stop(); stepper4.stop();
84         manualHoming();
85         Serial.println("DONE");
86         state = IDLE;
87         delay(300);
88         return;
89     }
90     // 2) Manual stepping state
91     if (state == MANUAL_STEP) {
92         if (manualID == 1) stepper1.run();
93         else if (manualID == 2) stepper2.run();
94         else if (manualID == 4) stepper4.run();
95         // when motor done
96         bool done = (manualID==1 && !stepper1.isRunning()) ||
97                     (manualID==2 && !stepper2.isRunning()) ||
98                     (manualID==4 && !stepper4.isRunning());
99         if (done) {
100             Serial.print("CURRENT,");
101             Serial.print(stepper1.currentPosition()); Serial.print(",");
102             Serial.print(stepper2.currentPosition()); Serial.print(",");
103             Serial.println(stepper4.currentPosition());
104             state = IDLE;
105         }
106         return;
107     }
108     // 3) Trajectory state with sequencing
109     if (state == TRAJECTORY) {
110         if (trajMode == 1) {
111             // simultaneous XY+Z
112             stepper1.run(); stepper2.run(); stepper3.run(); stepper4.run();
113             if (!stepper1.isRunning() && !stepper2.isRunning() &&
114                 !stepper3.isRunning() && !stepper4.isRunning()) {
115                 // now actuate gripper
116                 gripperServo.write(gripVal);
117                 delay(300);
118                 Serial.println("DONE");
119                 state = IDLE;
120             }
121         }
122     }
123     if (trajMode == 2) {
124         // Z -> then XY
125         if (trajPhase == PHASE1) {
126             stepper4.run();
127             if (!stepper4.isRunning()) {
128                 // start XY
129                 stepper1.moveTo(tarPos[0]);
130             }
131         }
132     }
133 }
```

ROBOT SCARA

GIAO DIỆN GUI



Screenshot of the Microsoft Visual Studio IDE showing the design view of a Windows Form application for a SCARA robot interface.

The application window title is "MyForm.h [Design]". The main form title is "MyForm". Inside the form:

- A logo featuring the Vietnamese flag colors (red, yellow, blue) with the text "ĐẠI HỌC BÁCH KHOA".
- Text labels: "NGUYỄN PHONG THÁI", "MSSV: 20226429", and "Môn học: Kỹ thuật lập trình Cơ Điện Tử".
- A large title "ROBOT SCARA" centered on the form.
- Input fields and spinners for coordinates X, Y, Z and joint angles J1, J2, Grip, Speed, Accel.
- Buttons: "Connect", "Calculate", "SAVE" (orange), "RUN" (green), "CLEAR", "HOME", and "Address".
- A numeric up-down control labeled "Steps: 0" with a slider and buttons "+Z" and "-Z".

The Solution Explorer shows the project structure:

- Solution 'GUISCARA20226429' (1)
- GUISCARA20226429
- References
- External Dependencies
- Header Files
- MyForm.h
- MyForm.resx
- Resource Files
- Source Files

The Properties panel shows settings for the "MyForm" class:

- BackgroundImage
- Cursor
- Font: Microsoft Sans
- ForeColor: ControlText
- FormBorderStyle: Fixed3D
- RightToLeft: No
- RightToLeftLayout: False
- Text: MyForm
- UseWaitCursor: False

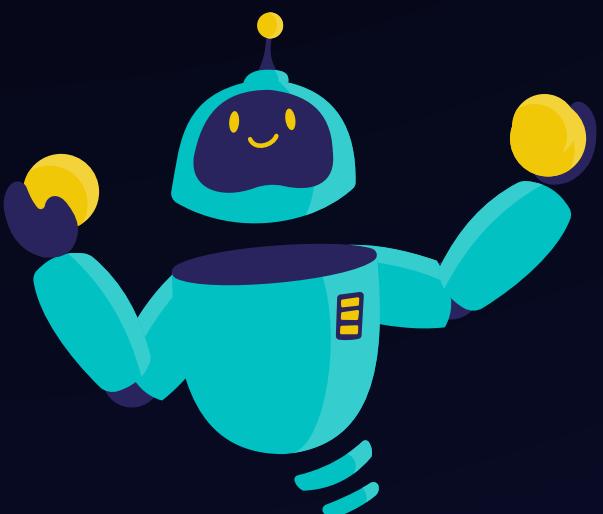
The Error List shows 0 Errors, 0 Warnings, and 0 Messages.

ROBOT SCARA GIAO DIỆN GUI

CHỨC NĂNG TỔNG QUAN

Điều khiển robot SCARA qua giao tiếp Serial với Arduino. Giao diện cho phép:

- Nhập tọa độ X, Y, Z → tính toán góc khớp J1, J2 (nghịch kinematics)
- Thiết lập tốc độ, gia tốc, gripper
- Chọn mode chạy (XYZ, Z-XY, XY-Z)
- Kết nối COM port, lưu lệnh, chạy lệnh, điều khiển thủ công từng bước



Thành phần	Vai trò
comboBoxPort	Chọn cổng COM để kết nối với Arduino
valueX, valueY, valueZ	Nhập tọa độ điểm đích
Theta1, Theta2	Góc robot sau khi tính toán
Grip, SpeedMotor, AccelMotor	Tham số điều khiển
Calculate	Tính toán góc J1, J2 từ X, Y bằng công thức IK
save	Lưu lệnh vào danh sách
run	Gửi toàn bộ danh sách lệnh cho Arduino
clear	Xóa lệnh khỏi danh sách
trackBarSteps	Điều chỉnh số bước khi di chuyển thủ công
up/downstep1/2, UpZ, DownZ	Điều khiển thủ công J1, J2, Z
Home	Gửi lệnh "HOMING" để về gốc
listBoxSteps	Hiển thị và thực thi lại các lệnh đã lưu

ROBOT SCARA GIAO DIỆN GUI

Giao tiếp Arduino

- Dùng SerialPort để gửi lệnh như:
 - "MANUAL,1,100": quay J1 thêm 100 bước
 - "1,1,90,45,0,50,90,1000,800,0": chạy theo tọa độ
 - "HOMING": đưa robot về gốc

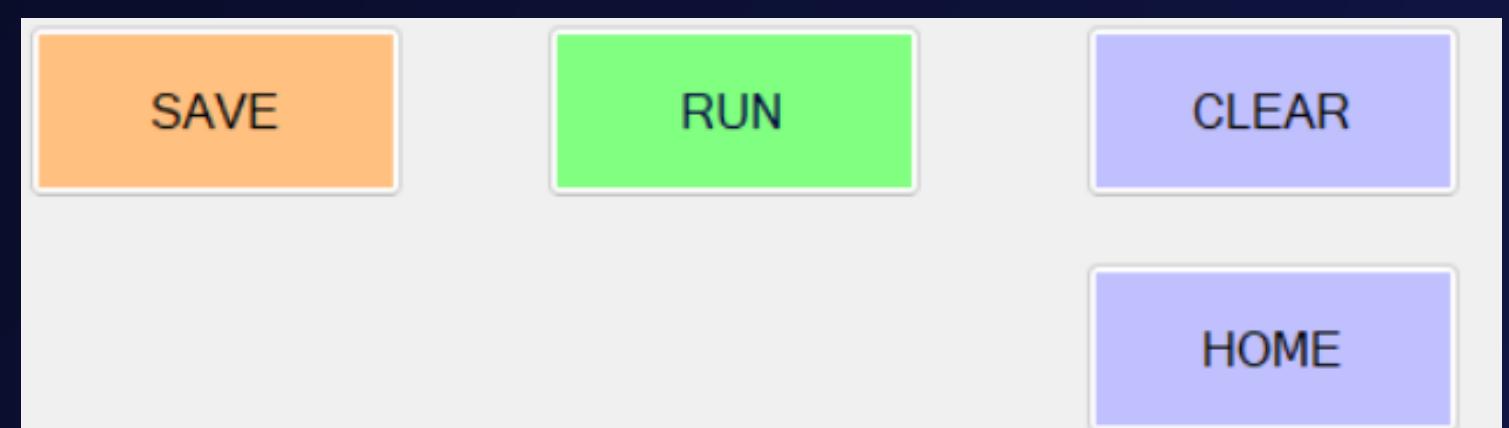
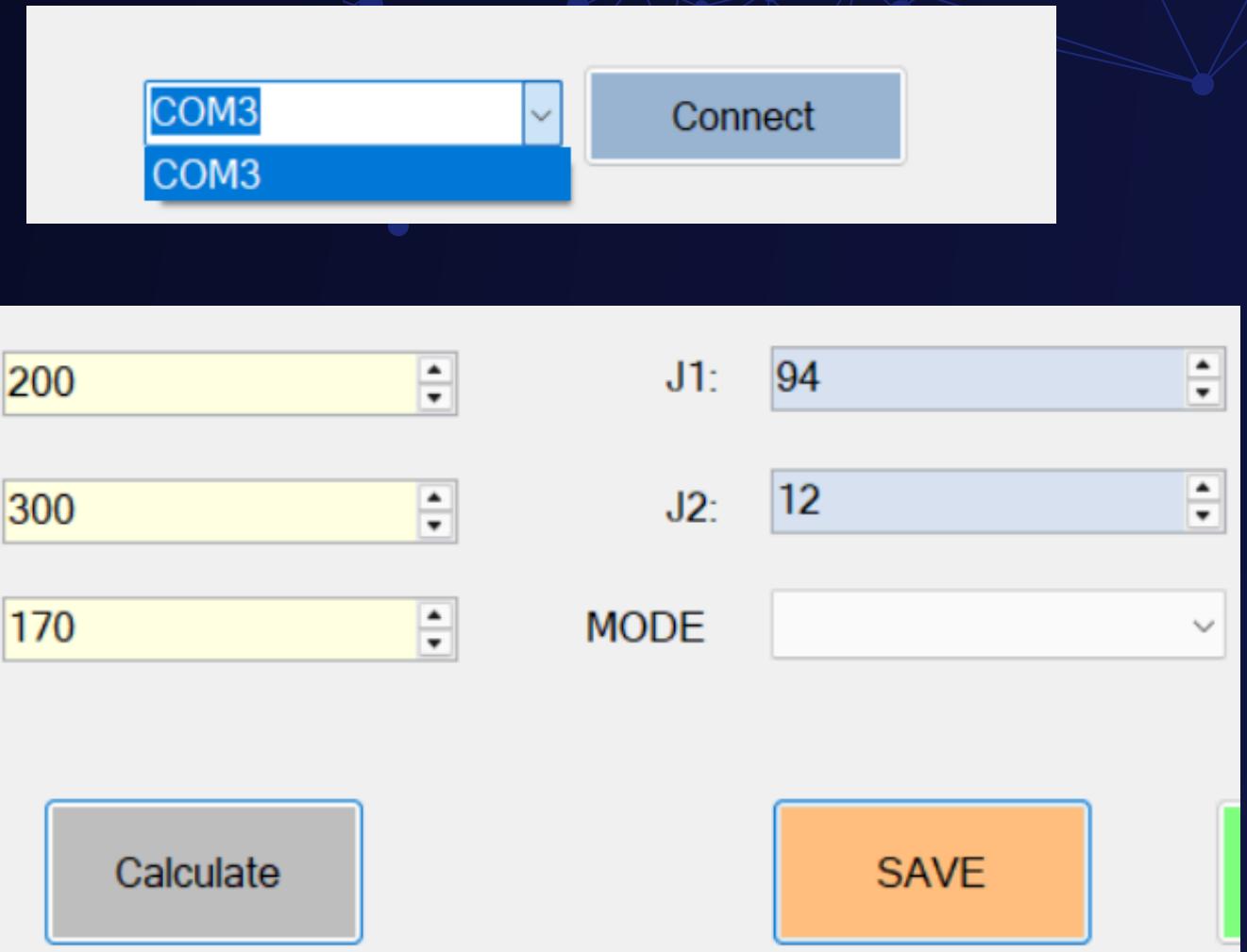
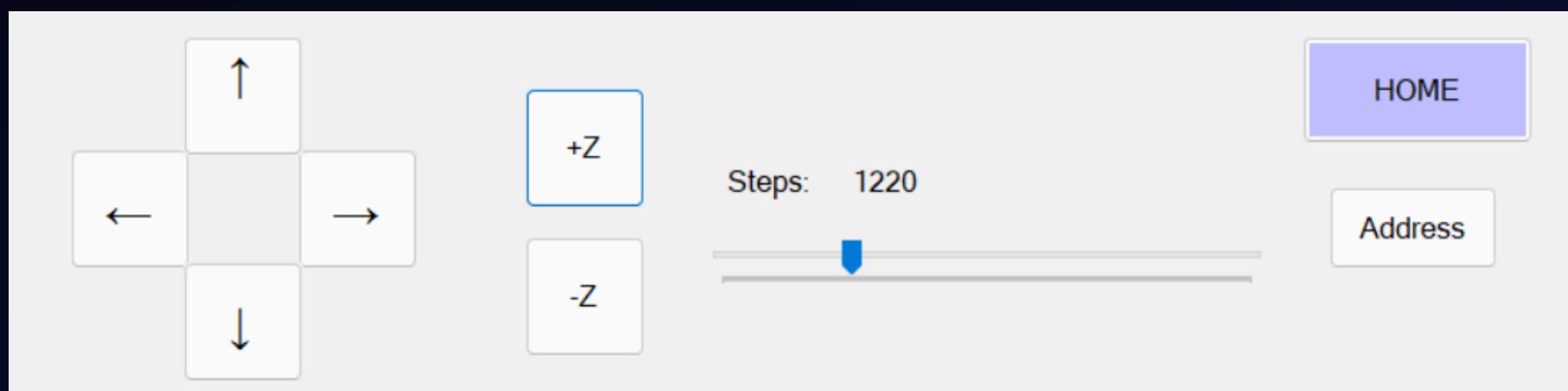
Thuật toán nghịch kinematics (IK)

Trong Calculate_Click():

Nhập X, Y → tính θ_1, θ_2 dựa theo công thức toán học của SCARA

Hiển thị kết quả lên GUI

Cụm điều khiển thủ công bằng nút bấm và thanh trượt để điều khiển góc xoay của khâu 1, khâu 2 và độ cao trực Z



ROBOT SCARA

GIAO DIỆN GUI



MyForm

ĐẠI HỌC
BÁCH KHOA

NGUYỄN PHONG THÁI
MSSV: 20226429
Môn học: Kỹ thuật lập trình Cơ Điện Tử

ROBOT SCARA

COM3 Connect

X: 200 J1: 94 Grip: 100
Y: 300 J2: 12 Speed: 2000
Z: 170 MODE XYZ Accel: 800

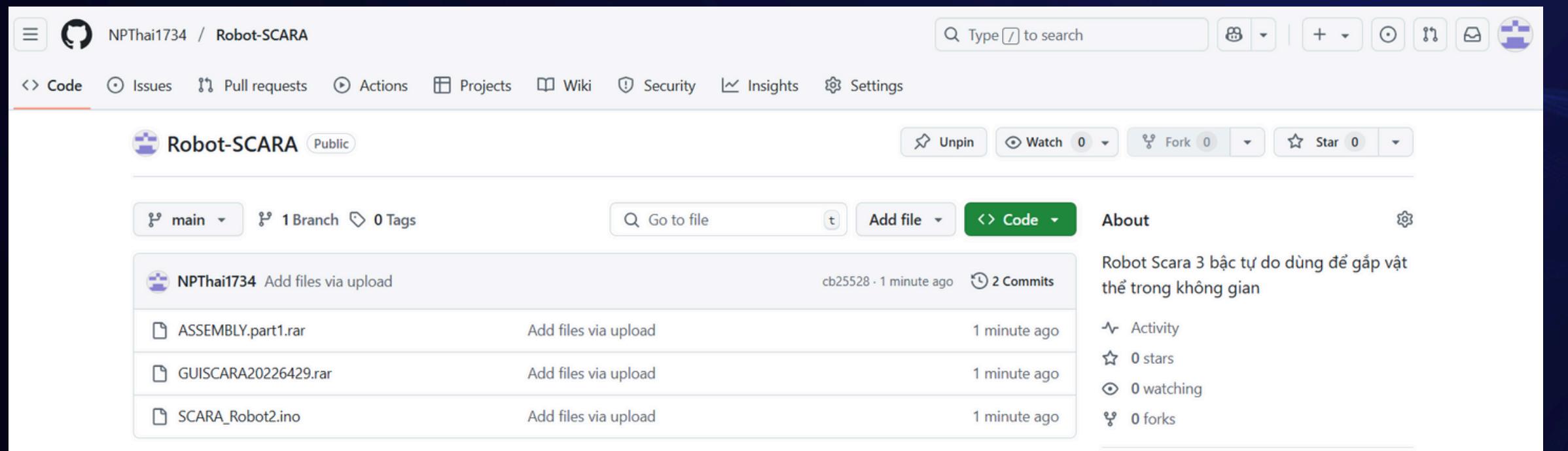
Calculate SAVE RUN CLEAR
HOME Address

+Z Steps: 1220 -Z

1,1,94,11,0,170,100,2000,800,0

Thank You

Thông tin dự án bài tập lớn ROBOT SCARA trên Github:
<https://github.com/NPThai1734/Robot-SCARA>



The screenshot shows the GitHub repository page for 'Robot-SCARA' owned by 'NPThai1734'. The repository is public and has 1 branch and 0 tags. It contains three files added via upload: 'ASSEMBLY.part1.rar', 'GUISCARA20226429.rar', and 'SCARA_Robot2.ino'. The last commit was made 1 minute ago by 'cb25528'. The repository has 0 stars, 0 forks, and 0 watching.

 0961001204

 thai.np226429@sis.hust.edu.vn

