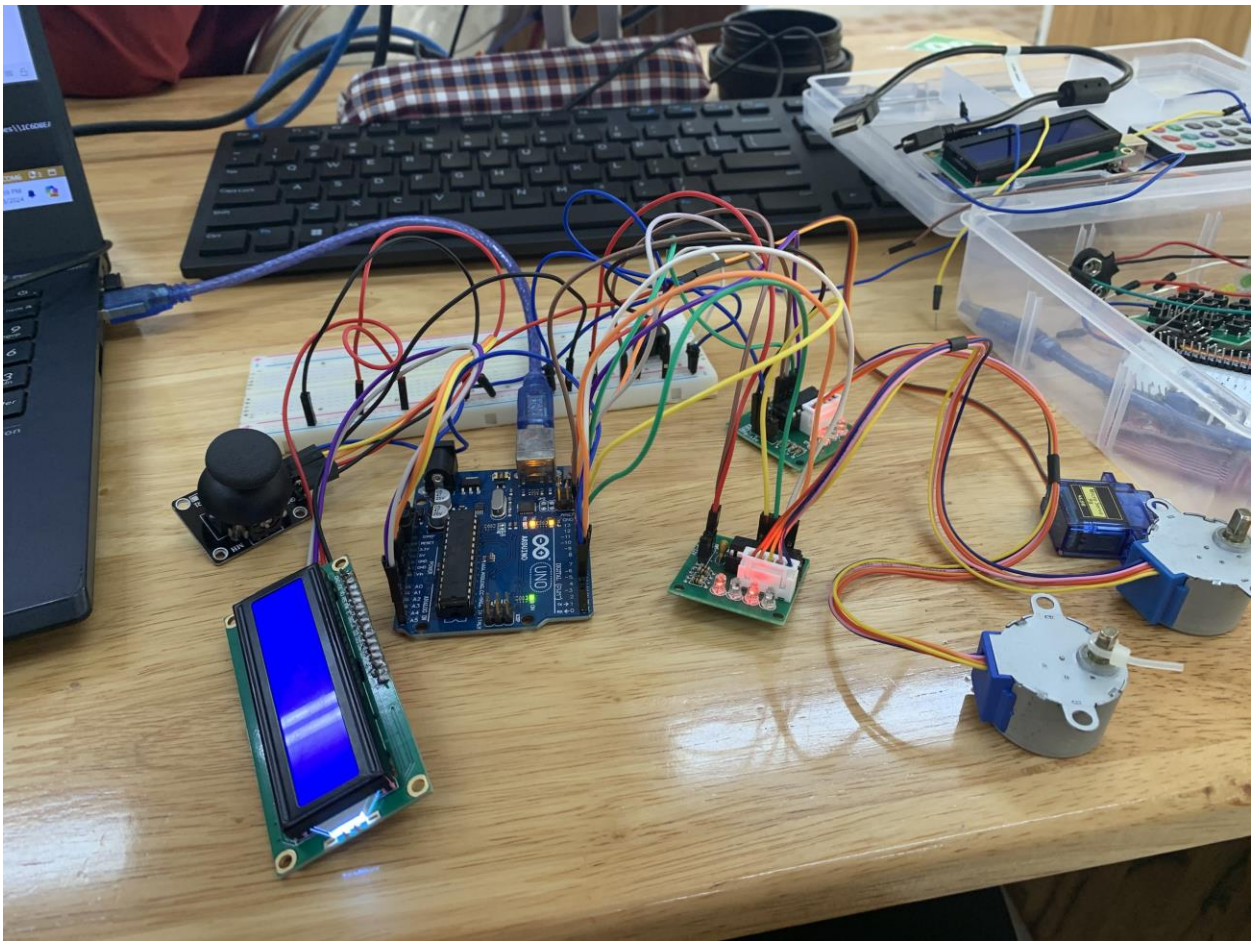

Bài 1. b. Lập trình điều khiển thay đổi hướng động cơ 28BYJ-48 khi điều khiển với Joystick?

```
#include <LiquidCrystal_I2C.h>
#include <Stepper.h>

// LCD
const int address = 0x27;
LiquidCrystal_I2C lcd(address, 16, 2);

const int stepsPerRevolution = 200;
const int motorPins[4] = {8, 9, 10, 11};

// Joystick pins
int bientroX = A0;
int bientroY = A1;

// Button (optional, not used in this example)
```

```
int button = 2;

int x = 0;
int y = 0;

Stepper myStepper(stepsPerRevolution, motorPins[0], motorPins[1], motorPins[2],
motorPins[3]);

void setup() {
  int i;
  for (i = 0; i < 4; i++) {
    pinMode(motorPins[i], OUTPUT);
  }
  pinMode(bientroX, INPUT);
  pinMode(bientroY, INPUT);
  Serial.begin(9600);

  // LCD initialization
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("X=");
  lcd.print(x);
  lcd.setCursor(0, 1);
  lcd.print("Y=");
  lcd.print(y);

  myStepper.setSpeed(100);
}

void loop() {
  x = analogRead(bientroX);
  y = analogRead(bientroY);
  Serial.print("X=");
  Serial.println(x);
  Serial.print("Y=");
  Serial.println(y);

  int direction = 0;
  if (y > 512) {
    direction = 1;
  } else if (y < 256) {
    direction = -1;
```

```
  }

  moveMyStepper(direction);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("X=");
  lcd.print(x);
  lcd.setCursor(0, 1);
  lcd.print("Y=");
  lcd.print(y);
  delay(500);
}

void moveMyStepper(int direction) {
  if (direction != 0) {
    int steps = map(abs(y - 512), 0, 512, 0, stepsPerRevolution);
    myStepper.step(direction * steps);
  } else {
    myStepper.step(0);
  }
}
```

Bài 1. c. Hiển thị góc xoay động cơ 28BYJ-48, chiều xoay, trạng thái lên màn hình LCD?

```
#include <LiquidCrystal_I2C.h>
#include <Stepper.h>

// LCD
const int address = 0x27;
LiquidCrystal_I2C lcd(address, 16, 2);

const int stepsPerRevolution = 200;
const int motorPins[4] = {8, 9, 10, 11};

// Joystick pins
int bientroX = A0;
int bientroY = A1;

int button = 2;
```

```
int x = 0;
int y = 0;

Stepper myStepper(stepsPerRevolution, motorPins[0], motorPins[1], motorPins[2],
motorPins[3]);

void setup() {
  int i;
  for (i = 0; i < 4; i++) {
    pinMode(motorPins[i], OUTPUT);
  }
  pinMode(bientroX, INPUT);
  pinMode(bientroY, INPUT);
  Serial.begin(9600);

  // LCD initialization
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("X=");
  lcd.print(x);
  lcd.setCursor(0, 1);
  lcd.print("Y=");
  lcd.print(y);

  myStepper.setSpeed(100);
}

void loop() {
  x = analogRead(bientroX);
  y = analogRead(bientroY);
  Serial.print("X=");
  Serial.println(x);
  Serial.print("Y=");
  Serial.println(y);

  int direction = 0;
  if (y > 512) {
    direction = 1;
  } else if (y < 256) {
    direction = -1;
  }
```

```
  moveMyStepper(direction);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("X=");
  lcd.print(x);
  lcd.setCursor(0, 1);
  lcd.print("Y=");
  lcd.print(y);
  delay(500);
}

void moveMyStepper(int direction) {
  if (direction != 0) {
    int steps = map(abs(y - 512), 0, 512, 0, stepsPerRevolution);
    myStepper.step(direction * steps);
  } else {
    myStepper.step(0);
  }
}
```

Bài 2.b Lập trình điều khiển thay đổi hướng động cơ Servo khi điều khiển với Joystick?

```
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

// LCD
const int address = 0x27;
LiquidCrystal_I2C lcd(address, 16, 2);

// Servo pin
const int servoPin = 12;
Servo myServo;

// Joystick pins
int bientroX = A0;
int bientroY = A1;

int button = 2;

int x = 0;
```

```
int y = 0;

void setup() {
  pinMode(bientroX, INPUT);
  pinMode(bientroY, INPUT);
  Serial.begin(9600);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("X=");
  lcd.print(x);
  lcd.setCursor(0, 1);
  lcd.print("Y=");
  lcd.print(y);

  myServo.attach(servoPin);
  myServo.write(0);
}

void loop() {
  x = analogRead(bientroX);
  y = analogRead(bientroY);
  Serial.print("X=");
  Serial.println(x);
  Serial.print("Y=");
  Serial.println(y);

  int servoPos = map(y, 0, 1023, 0, 180);

  moveMyServo(servoPos);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("X=");
  lcd.print(x);
  lcd.setCursor(0, 1);
  lcd.print("Y=");
  lcd.print(y);
  delay(500);
}

void moveMyServo(int pos) {
```

```
  myServo.write(pos);
}
```

Bài 2.c Nhấn nhấn nút giữa (SW) xoay động cơ Servo về vị trí 0 độ?

```
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

// LCD
const int address = 0x27;
LiquidCrystal_I2C lcd(address, 16, 2);

// Servo pin
const int servoPin = 12;
Servo myServo;

// Joystick pins
int bientroX = A0;
int bientroY = A1;

// Button pin
int button = 2;

int x = 0;
int y = 0;

void setup() {
  pinMode(bientroX, INPUT);
  pinMode(bientroY, INPUT);
  pinMode(button, INPUT_PULLUP);
  Serial.begin(9600);

  // LCD initialization
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("X=");
  lcd.print(x);
  lcd.setCursor(0, 1);
  lcd.print("Y=");
  lcd.print(y);
```

```
  myServo.attach(servoPin);
  myServo.write(0);
}

void loop() {
  x = analogRead(bientroX);
  y = analogRead(bientroY);
  Serial.print("X=");
  Serial.println(x);
  Serial.print("Y=");
  Serial.println(y);

  if (digitalRead(button) == LOW) {
    // myServo.write(0);
  } else {
    int servoPos = map(y, 0, 1023, 0, 180);

    moveMyServo(servoPos);
  }

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("X=");
  lcd.print(x);
  lcd.setCursor(0, 1);
  lcd.print("Y=");
  lcd.print(y);
  delay(500);
}

void moveMyServo(int pos) {
  myServo.write(pos);
}
```

Bài 3: Lập trình điều khiển thay đổi hướng 2 động cơ 28BYJ-48 khi điều khiển với Joystick?

- Joystick lên và xuống: Điều khiển động cơ 28BYJ-48 thứ nhất?

- Joystick trái và phải: Điều khiển động cơ 28BYJ-48 thứ hai?

- Kết hợp trái và phải và lên và xuống để điều khiển cả hai động cơ 28BYJ-48?

```
#include <LiquidCrystal_I2C.h>
#include <Stepper.h>

// LCD
const int address = 0x27; // I2C address of LCD (default 16x2)
LiquidCrystal_I2C lcd(address, 16, 2);

// Stepper motor configuration (adjust based on your motor and driver)
const int stepsPerRevolution = 2048; // Replace with your motor's steps per
revolution
const int motorPins1[4] = {8, 9, 10, 11}; // Replace with your motor driver pins
for motor 1
const int motorPins2[4] = {4, 5, 6, 7}; // Replace with your motor driver pins
for motor 2

// Joystick pins
int bientroX = A0;
int bientroY = A1;

// Button pin
int button = 2;

int x = 0;
int y = 0;

Stepper myStepper1(stepsPerRevolution, motorPins1[0], motorPins1[1],
motorPins1[2], motorPins1[3]);
Stepper myStepper2(stepsPerRevolution, motorPins2[0], motorPins2[1],
motorPins2[2], motorPins2[3]);

void setup() {
  int i;
  for (i = 0; i < 4; i++) {
    pinMode(motorPins1[i], OUTPUT); // Set motor driver pins for motor 1 as
output
    pinMode(motorPins2[i], OUTPUT); // Set motor driver pins for motor 2 as
output
  }
  pinMode(bientroX, INPUT);
  pinMode(bientroY, INPUT);
  pinMode(button, INPUT_PULLUP); // Set button pin as input with internal pull-up
resistor
  Serial.begin(9600);
```

```
  // LCD initialization
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("X=");
  lcd.print(x);
  lcd.setCursor(0, 1);
  lcd.print("Y=");
  lcd.print(y);

  // Set stepper motor speed (adjust as needed)
  myStepper1.setSpeed(10000); // Steps per second for motor 1
  myStepper2.setSpeed(10000); // Steps per second for motor 2
}

void loop() {
  x = analogRead(bientroX);
  y = analogRead(bientroY);
  // Serial.print("X=");
  // Serial.println(x);
  // Serial.print("Y=");
  // Serial.println(y);

  // Determine motor directions and speed based on joystick X and Y axes
  int direction1 = 0; // 0: Stop, 1: CW, -1: CCW for motor 1
  int direction2 = 0; // 0: Stop, 1: CW, -1: CCW for motor 2
  int speed1 = map(abs(y - 512), 0, 512, 0, 100); // Map Y-axis value to motor
speed (0-100%)
  int speed2 = map(abs(x - 512), 0, 512, 0, 100); // Map X-axis value to motor
speed (0-100%)

  if (y > 700) {
    direction1 = 1; // Up (CW for motor 1)
  } else if (y < 300) {
    direction1 = -1; // Down (CCW for motor 1)
  }

  if (x > 700) {
    direction2 = 1; // Right (CW for motor 2)
  } else if (x < 300) {
    direction2 = -1; // Left (CCW for motor 2)
  }
```

```
  // Control motors using moveMySteppers function
  moveMySteppers(direction1, direction2, speed1, speed2);

  // Update LCD and Serial monitor
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("X=");
  lcd.print(x);
  lcd.setCursor(0, 1);
  lcd.print("Y=");
  lcd.print(y);
  delay(100);
}

// Function to move the stepper motors
void moveMySteppers(int direction1, int direction2, int speed1, int speed2) {
  if (direction1 != 0) {
    int steps1 = map(speed1, 0, 100, 0, stepsPerRevolution); // Map speed
percentage to number of steps
    myStepper1.step(direction1 * steps1);
  }

  if (direction2 != 0) {
    int steps2 = map(speed2, 0, 100, 0, stepsPerRevolution); // Map speed
percentage to number of steps
    myStepper2.step(direction2 * steps2);
  }
}
```