

```

// --- Definições de Hardware (CNC Shield - Eixo X) ---
#define STEP_PIN      2
#define DIR_PIN       5
#define ENABLE_PIN    8

// --- Definição de Parâmetros Mecânicos ---
// O motor tem 200 passos/volta (1.8 graus/passo).
// Com driver em full-step: 200 passos = 360 graus.
// Se estiver a usar microstepping (ex: 1/16), estes valores devem ser ajustados.
const int PASSOS_PARA_90_GRAUS = 800;    // Ajustar conforme o microstepping
configurado nos jumpers
const int PASSOS_PARA_180_GRAUS = 1600;

// --- Variáveis de Estado --- (FIM DE CURSO POR IMPLEMENTAR)
// 0 = Centro (0 graus)
// 1 = Esquerda (+90 graus)
// 2 = Direita (-90 graus)
int posicaoAtual = 0;

void setup() {
    // Configuração dos pinos do driver A4988
    pinMode(STEP_PIN, OUTPUT);
    pinMode(DIR_PIN, OUTPUT);
    pinMode(ENABLE_PIN, OUTPUT);

    // ATIVAÇÃO DO MOTOR:
    // O pino Enable é ativo a LOW. Ao colocar LOW, as bobinas são energizadas
    // e o motor ganha binário de retenção (fica duro).
    digitalWrite(ENABLE_PIN, LOW);

    // Inicia a comunicação Série (UART) com o Arduino MEGA
    Serial.begin(9600);
}

void loop() {
    // Verifica se o Arduino MEGA enviou algum comando
    if (Serial.available() > 0) {
        char comando = Serial.read();

        // --- MÁQUINA DE ESTADOS PARA CONTROLO DE POSIÇÃO ---

        // CASO 1: Comando para ir ao CENTRO (0)
        if (comando == '0' && posicaoAtual != 0) {
            if (posicaoAtual == 1) { // Estava em +90
                moveMotor(PASSOS_PARA_90_GRAUS, LOW); // Recua 90
            }
            else if (posicaoAtual == 2) { // Estava em -90
                moveMotor(PASSOS_PARA_90_GRAUS, HIGH); // Avança 90
            }
        }
    }
}

```

```

    posicaoAtual = 0; // Atualiza estado
}

// CASO 2: Comando para ir para ESQUERDA (+90)
else if (comando == '1' && posicaoAtual != 1) {
    if (posicaoAtual == 0) { // Estava em 0
        moveMotor(PASSOS_PARA_90_GRAUS, HIGH); // Avança 90
    }
    else if (posicaoAtual == 2) { // Estava em -90
        moveMotor(PASSOS_PARA_180_GRAUS, HIGH); // Avança 180 (atravessa o centro)
    }
    posicaoAtual = 1;
}

// CASO 3: Comando para ir para DIREITA (-90)
else if (comando == '2' && posicaoAtual != 2) {
    if (posicaoAtual == 0) { // Estava em 0
        moveMotor(PASSOS_PARA_90_GRAUS, LOW); // Recua 90
    }
    else if (posicaoAtual == 1) { // Estava em +90
        moveMotor(PASSOS_PARA_180_GRAUS, LOW); // Recua 180
    }
    posicaoAtual = 2;
}
}

// Função auxiliar para gerar os pulsos de passo
void moveMotor(int numPassos, bool direcao) {
    digitalWrite(DIR_PIN, direcao);

    for (int x = 0; x < numPassos; x++) {
        digitalWrite(STEP_PIN, HIGH);
        delayMicroseconds(500); // Define a velocidade do motor
        digitalWrite(STEP_PIN, LOW);
        delayMicroseconds(500);
    }
}

```