if (Serial.available() > 0) {

char command = Serial.read();

float distance = 5;

switch (command) {

case 'X':

bool forward = TRUE;

moveX(stepPin1, stepPin2, distance, forward)// Code to move X motor forward

break;

case 'x':

bool forward = FALSE;

moveX(stepPin1, stepPin2, distance, forward)// Code to move X motor backward

break;

case 'Y':

bool forward = TRUE;

moveY(int stepPin, int steps)// Code to move Y motor forward

break;

case 'y':

bool forward = FALSE;

moveY(int stepPin, int steps)// Code to move Y motor backward

break;

case 'Z':

bool forward = TRUE;

moveZ(int stepPin, int steps)// Code to move Z motor forward

break;

case 'z':

bool forward = FALSE;

moveZ(int stepPin, int steps)// Code to move Z motor backward

break;

case's1':

int speed = 10000;

break

case's2':

int speed = 1000;

break

case's3':

int speed = 100;

break

default:

break;

}

}

}

temperatureFanControl();

lightControl();

}

// Function to move both stepper motors of the X-axis for a given distance and direction

void moveX(int stepPin1, int stepPin2, float distance, bool forward, int speed) {

// Calculate the number of steps required based on the distance

float stepsPerMM = 2000; // 200 steps per revolution, and 10 revolutions per mm (0.1mm per revolution)

int steps = distance \* stepsPerMM;

// Set the direction of movement based on the 'forward' parameter

digitalWrite(dirPinX1, forward ? LOW : HIGH); // Assuming dirPinX1 is one direction pin for X-axis

digitalWrite(dirPinX2, forward ? LOW : HIGH); // Assuming dirPinX2 is the other direction pin for X-axis

// Loop through the calculated number of steps

for (int i = 0; i < steps; i++) {

// Trigger an impulse to move the motors

digitalWrite(stepPin1, HIGH);

digitalWrite(stepPin2, HIGH);

delayMicroseconds(speed); // Adjust delay as needed for motor speed

digitalWrite(stepPin1, LOW);

digitalWrite(stepPin2, LOW);

delayMicroseconds(speed); // Adjust delay as needed for motor speed

}

}

// Function to move the stepper motor of the Y-axis for a given distance and direction

void moveY(int stepPin, float distance, bool forward) {

// Calculate the number of steps required based on the distance

float stepsPerMM = 2000; // 200 steps per revolution, and 10 revolutions per mm (0.1mm per revolution)

int steps = distance \* stepsPerMM;

// Set the direction of movement based on the 'forward' parameter

digitalWrite(dirPinY, forward ? LOW : HIGH); // Assuming dirPinY is the direction pin for Y-axis

// Loop through the calculated number of steps

for (int i = 0; i < steps; i++) {

// Trigger an impulse to move the motor

digitalWrite(stepPin, HIGH);

delayMicroseconds(speed); // Adjust delay as needed for motor speed

digitalWrite(stepPin, LOW);

delayMicroseconds(speed); // Adjust delay as needed for motor speed

}

}

// Function to move the stepper motor of the Z-axis for a given distance and direction

void moveZ(int stepPin, float distance, bool forward) {

// Calculate the number of steps required based on the distance

float stepsPerMM = 2000; // 200 steps per revolution, and 10 revolutions per mm (0.1mm per revolution)

int steps = distance \* stepsPerMM;

// Set the direction of movement based on the 'forward' parameter

digitalWrite(dirPinZ, forward ? LOW : HIGH); // Assuming dirPinZ is the direction pin for Z-axis

// Loop through the calculated number of steps

for (int i = 0; i < steps; i++) {

// Trigger an impulse to move the motor

digitalWrite(stepPin, HIGH);

delayMicroseconds(speed); // Adjust delay as needed for motor speed

digitalWrite(stepPin, LOW);

delayMicroseconds(speed); // Adjust delay as needed for motor speed

}

}