**Smart Compost System Arduino Code**

Steps  
1. Add the sensor with its components in the mixture of compost

2. Attach the Arduino uno to power supply(laptop)

3. When compost is ready, the buzzer beeps, and the LED turns on with the sensor readings displayed on LCD.

4. At least when the compost is ready, the message "compost ready" is displayed on the LCD screen.

**Code**

#include <LiquidCrystal.h>

#include <Servo.h>

// Initialize the LCD (RS, E, DB4, DB5, DB6, DB7)

LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

// Define pins for components

#define MOISTURE\_SENSOR A0

#define TEMP\_SENSOR A1       // Simulated analog pin for temperature sensor

#define PH\_SENSOR A2         // Simulated analog pin for pH sensor

#define BUZZER\_PIN 13

#define TEMP\_LED\_PIN 2

#define MOIST\_LED\_PIN 3

#define MOTOR\_EN 5           // Motor Driver Enable Pin

#define MOTOR\_IN1 4          // Motor Driver Input 1

#define MOTOR\_IN2 6          // Motor Driver Input 2

#define SERVO\_PIN 9          // Servo Motor Pin

Servo servoMotor;            // Servo motor object

bool compostReady = false;   // Flag to track compost readiness

void setup() {

  // Initialize LCD

  lcd.begin(16, 2); // 16x2 LCD

  lcd.print("System Initializing");

  delay(2000); // Display initialization message for 2 seconds

  lcd.clear();

  // Setup pins

  pinMode(MOISTURE\_SENSOR, INPUT);

  pinMode(TEMP\_SENSOR, INPUT);

  pinMode(PH\_SENSOR, INPUT);

  pinMode(BUZZER\_PIN, OUTPUT);

  pinMode(TEMP\_LED\_PIN, OUTPUT);

  pinMode(MOIST\_LED\_PIN, OUTPUT);

  pinMode(MOTOR\_EN, OUTPUT);

  pinMode(MOTOR\_IN1, OUTPUT);

  pinMode(MOTOR\_IN2, OUTPUT);

  // Initialize Servo

  servoMotor.attach(SERVO\_PIN);

  servoMotor.write(0); // Move servo to initial position

  // Initialize LCD display

  lcd.print("System Reading");

  delay(2000);

  lcd.clear();

  Serial.begin(9600); // For debugging

}

void loop() {

  // Read sensors

  digitalWrite(BUZZER\_PIN, HIGH);

  int moistureLevel = analogRead(MOISTURE\_SENSOR);

  int tempLevel =analogRead(TEMP\_SENSOR); // Simulated temperature sensor

  int pHLevel = analogRead(PH\_SENSOR);

  // Simulated temperature conversion (adjust as needed)

  int temperature = map(tempLevel, 0, 1023, -10, 50); // Example: -10°C to 50°C

  float pH = (pHLevel / 1023.0) \* 14.0; // Example: Map analog value to 0-14 pH scale

  digitalWrite(BUZZER\_PIN, HIGH);

  // Determine compost readiness condition

  if (moistureLevel > 700 && temperature > 25 && temperature < 40 && pH > 11 && pH < 14) {

    compostReady = true; // Compost is ready

    digitalWrite(TEMP\_LED\_PIN, HIGH); // Turn on LED to indicate readiness

    digitalWrite(MOIST\_LED\_PIN, HIGH);

  } else {

    compostReady = false; // Compost is not ready

    digitalWrite(TEMP\_LED\_PIN, LOW);

    digitalWrite(MOIST\_LED\_PIN, LOW);

  }

  // Display sensor values on LCD

  lcd.setCursor(0, 0);

  lcd.print("Temp: ");

  lcd.print(temperature);

  lcd.print("C");

  lcd.setCursor(0, 1);

  lcd.print("Moist: ");

  lcd.print(moistureLevel);

  delay(5000);

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("pH: ");

  lcd.print(pH);

  lcd.setCursor(0, 1);

  lcd.print("Moist: ");

  lcd.print(moistureLevel);

  delay(2000);

  lcd.clear();

  // Control Buzzer

  if (compostReady) {

    lcd.print("Compost Ready");

    digitalWrite(BUZZER\_PIN, HIGH); // Turn on buzzer

    delay(10000); // Keep buzzer on for 30 seconds

    digitalWrite(BUZZER\_PIN, LOW); // Turn off buzzer

    compostReady = false; // Reset compostReady flag to avoid repeated buzzing

  }

  // Control Servo Motor (Simulate blade movement)

  if (moistureLevel < 300) { // Example: Blade activation condition

    servoMotor.write(90); // Rotate servo to 90 degrees

    delay(2000);

    servoMotor.write(0); // Return to initial position

  }

  // Control Motor (Simulate a motor-driven component, e.g., compost mixing)

  if (pH < 6 || pH > 8) { // Example: Motor activation condition

    digitalWrite(MOTOR\_EN, HIGH); // Enable motor

    digitalWrite(MOTOR\_IN1, HIGH); // Rotate motor in one direction

    digitalWrite(MOTOR\_IN2, LOW);

    delay(10000); // Run motor for 5 seconds

    digitalWrite(MOTOR\_EN, LOW); // Turn off motor

  }

}