### **Introduction**

This guide provides detailed instructions to build an open-source automated compost machine. It combines Arduino with additional sensors and an optional ESP32 module for wireless functionality. Follow the steps closely to replicate this project.

## **Materials Required**

### **Electronics**

1. Arduino UNO R3 (1x)
2. DC Motor (2x)
3. Ultrasonic Sensor (1x)
4. Water Level Sensor (1x)
5. IR Receiver (1x)
6. Servo Motor (1x)
7. 16x2 LCD Display with I2C Module (1x)
8. RGB LED Module (1x)
9. ESP32 Module (Optional, 1x)
10. Breadboard (1x)
11. Jumper Wires (Male-to-Male and Male-to-Female)
12. Resistors (220Ω for RGB and LCD)
13. Potentiometer (1x for LCD contrast adjustment)

### **Mechanical Components**

1. Compost Bin (Plastic/Metal)
2. Wooden/Metal Frame (to mount components)
3. Screws and Screwdrivers
4. 3D-Printed Motor Mounts (Optional, can also be purchased)

### **Power Supply**

1. 9V Battery or USB Cable (to power the Arduino)
2. Battery Holder (if using a battery)
3. Optional: DC power adapter (for continuous operation).

## **1. Setting Up the Arduino**

### **Step 1.1: Download Arduino IDE**

* Visit the official Arduino website and download the IDE software compatible with your operating system.
* Install the IDE and ensure it is functional by opening the program.

### **Step 1.2: Install Required Libraries**

* Go to **Tools > Manage Libraries** in the Arduino IDE.
* Search and install the following libraries:
  + LiquidCrystal for the LCD screen.
  + Servo for servo motor control.
  + NewPing for ultrasonic sensor operation.

### **Step 1.3: Test the Arduino Board**

* Connect the Arduino to your computer using a USB cable.
* Open the IDE and upload a basic "Blink" sketch:
  + Go to **File > Examples > Basics > Blink**.
  + Ensure the correct port and board are selected under **Tools**.

## **2. Connecting the Components**

### **Step 2.1: Wiring the LCD Display**

1. Connect the **VSS** pin of the LCD to GND.
2. Connect the **VDD** pin to 5V.
3. Adjust contrast using the **Vo** pin connected to a potentiometer.
4. Connect the **Rs** pin to digital pin 8.
5. Connect **E** to digital pin 9.
6. Connect **D4-D7** to digital pins D4-D7 on the Arduino.
7. Connect **A (Anode)** and **K (Cathode)** for backlight to 5V and GND, respectively, with a resistor for the anode.

### **Step 2.2: Wiring the DC Motor Driver**

1. Connect the **Enable** pin of the motor driver to digital pin 11.
2. Connect the **Diva** and **Dirb** pins to digital pins 12 and 13, respectively.
3. Power the motor driver by connecting its **VSS** pin to 5V and **GND** to GND.
4. Wire the output terminals of the motor driver to the DC motor leads.

### **Continuing...**

## **2. Connecting the Components (Continued)**

### **Step 2.3: Wiring the Ultrasonic Sensor**

1. Connect the **VSS** pin to 5V.
2. Connect the **GND** pin to GND.
3. Attach the **Trig** pin to **A2** on the Arduino.
4. Attach the **Echo** pin to **A1**.

### **Step 2.4: Wiring the Servo Motor**

1. Connect the signal wire (yellow) to digital pin **10**.
2. Connect the VSS wire (red) to 5V.
3. Connect the GND wire (brown) to GND.

### **Step 2.5: Wiring the Water Level Sensor**

1. Connect the signal pin (S) to analog pin **A0**.
2. Connect the VSS pin to 5V.
3. Connect the GND pin to GND.

### **Step 2.6: Wiring the IR Receiver**

1. Connect the **OUT** pin to digital pin **3**.
2. Connect the **VSS** pin to 5V.
3. Connect the **GND** pin to GND.

## **3. Optional ESP32 Module Setup**

### **Step 3.1: Preparing the ESP32**

1. Ensure the Arduino IDE is set up for ESP32:
   * Go to **File > Preferences > Additional Board Manager URLs**.
   * Add the URL for ESP32 boards (find it on the official ESP32 GitHub page).
   * Install the ESP32 board from **Tools > Board > Board Manager**.

### **Step 3.2: ESP32 Wiring**

1. Connect **3.3V (EN)** to the ESP32's **VIN**.
2. Connect the **GND** pin to GND.
3. Connect **TX** of ESP32 to **RX** on Arduino.
4. Connect **RX** of ESP32 to **TX** on Arduino.
   * Note: Use level shifters if Arduino is 5V logic and ESP32 is 3.3V logic.

### **Step 3.3: Upload the ESP32 Program**

* Write a basic sketch to enable Wi-Fi or Bluetooth functionality.
* Example:

#include <WiFi.h>

void setup() {

WiFi.begin("SSID", "password");

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

}

}

void loop() {

// Your code for Wi-Fi functionality

}

## **4. Programming the Arduino**

### **Step 4.1: Writing the Arduino Code**

1. Open a new sketch in the Arduino IDE.
2. Copy and paste the complete code for the project, ensuring proper pin definitions for each component.
3. Define libraries and include initialization code for each sensor. Example:

#include <LiquidCrystal.h>

#include <Servo.h>

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

Servo servoMotor;

void setup() {

lcd.begin(16, 2);

servoMotor.attach(10);

// Additional setups

}

### **Continuing...**

## **4. Programming the Arduino (Continued)**

### **Step 4.2: Integrating Sensors and Motors**

1. Write functions to handle each sensor and component:  
   * **Ultrasonic Sensor**: Measure distance.

long measureDistance() {

digitalWrite(A2, LOW);

delayMicroseconds(2);

digitalWrite(A2, HIGH);

delayMicroseconds(10);

digitalWrite(A2, LOW);

long duration = pulseIn(A1, HIGH);

long distance = duration \* 0.034 / 2; // Convert to cm

return distance;

}

* + **Water Level Sensor**: Read analog values.

int waterLevel = analogRead(A0);

* + **Servo Motor**: Open/close the compost bin.

void openLid() {

servoMotor.write(90); // Adjust angle for your bin

delay(1000);

servoMotor.write(0); // Close after delay

}

1. Add logic for decision-making:  
   * Example: Open the lid when waste is detected within 10 cm.

if (measureDistance() < 10) {

openLid();

}

1. Program the RGB LED for status indication:  
   * Green for "Ready," Red for "Error," Blue for "Processing."

void setLED(String status) {

if (status == "Ready") {

digitalWrite(A1, HIGH);

digitalWrite(A2, LOW);

digitalWrite(A3, LOW);

} else if (status == "Error") {

digitalWrite(A1, LOW);

digitalWrite(A2, HIGH);

digitalWrite(A3, LOW);

} else if (status == "Processing") {

digitalWrite(A1, LOW);

digitalWrite(A2, LOW);

digitalWrite(A3, HIGH);

}

}

## **5. Assembling the Hardware**

### **Step 5.1: Mounting the Components**

1. Secure the compost bin to the frame.
2. Mount the DC motors at the base for mixing.
3. Place the ultrasonic sensor at the top of the bin to measure waste levels.
4. Attach the servo motor to the bin lid for automatic opening and closing.
5. Fix the water level sensor inside the bin near the base.

### **Step 5.2: Organizing Wires**

* Use zip ties or adhesive clips to manage wires neatly.
* Keep power and data wires separate to avoid interference.

### **Step 5.3: Testing the Wiring**

1. Power on the system.
2. Check the LCD for initialization messages.
3. Verify the RGB LED status changes based on input.
4. Test each motor and sensor independently.

## **6. Testing the System**

### **Step 6.1: Initial Run**

1. Place biodegradable waste inside the bin.
2. Observe the ultrasonic sensor activating the motor when waste is detected.
3. Check the water level sensor for moisture levels.
4. Monitor the LCD for updates (e.g., "Processing Compost," "Ready").

### **Step 6.2: Debugging**

* If the system doesn’t respond as expected:
  + Recheck wiring connections.
  + Use the Arduino Serial Monitor to debug sensor readings.

### **Continuing...**

## **7. Optional Features**

### **Step 7.1: Adding ESP32 for Wireless Control**

If you choose to incorporate the ESP32 module:

1. Install the ESP32 library and drivers as mentioned earlier.
2. Program the ESP32 to connect to Wi-Fi and communicate with a mobile app (e.g., Glide).
3. Set up a simple protocol for commands (e.g., turn on/off motors remotely).

Example Arduino code for communication with ESP32 via Serial:

void loop() {

if (Serial.available()) {

char command = Serial.read();

if (command == 'O') { // Open lid

openLid();

} else if (command == 'M') { // Start motor

digitalWrite(11, HIGH); // Enable motor

} else if (command == 'S') { // Stop motor

digitalWrite(11, LOW);

}

}

}

### **Step 7.2: Real-Time Monitoring**

* Use the ESP32 to send data (e.g., water level, waste status) to a cloud platform or a mobile app for monitoring.
* This enhances user interaction and allows remote operation.

## **8. Troubleshooting**

### **Step 8.1: Common Issues**

1. **Sensor Malfunction**:  
   * Verify connections and replace damaged sensors.
   * Use a multimeter to check power supply consistency.
2. **Motor Not Running**:  
   * Check the motor driver wiring and ensure it’s receiving power.
   * Confirm PWM signals from the Arduino.
3. **LCD Display Not Working**:  
   * Adjust the potentiometer for contrast.
   * Ensure proper connections to pins 4–7 on the Arduino.

### **Step 8.2: Debugging Tips**

* Use the Arduino Serial Monitor to log sensor readings and outputs.
* Test components independently before integrating them into the full system.

## **9. Maintaining Your Compost Machine**

### **Step 9.1: Cleaning**

* Regularly clean the bin to prevent waste buildup.
* Wipe sensors to ensure accurate readings.

### **Step 9.2: Component Care**

* Keep the motor free from debris.
* Avoid exposing electronic components to excessive moisture.

### **Step 9.3: Updates**

* Periodically update the Arduino code to improve functionality or add new features.

## **10. Final Notes**

Congratulations! You’ve successfully built an open-source automated compost machine. By following these steps, you can contribute to waste management while learning valuable electronics and programming skills.

For further details, consult the repository’s README or reach out to the open-source community!