

Source Code

```
#include <Wire.h>

#include <RTCLib.h>

#include "HX711.h"

RTC_DS3231 rtc;      // Real-time clock object


const int triggerPin = 9; // Ultrasonic sensor trigger pin
const int echoPin = 10;  // Ultrasonic sensor echo pin


const int weightDataPin = A1; // Weight sensor data pin
const int weightClockPin = A2; // Weight sensor clock pin
const int methaneSensorPin = A3; // Methane gas sensor analog pin
const int co2SensorPin = A4;  // CO2 gas sensor analog pin


HX711 scale; // Weight sensor object


const int thresholdDistance = 20; // Threshold distance for ultrasonic sensor (in cm)
const float thresholdWeight = 100.0; // Threshold weight for weight sensor (in grams)
const int thresholdMethaneValue = 500; // Threshold methane gas sensor value (adjust as needed)
const int thresholdCO2Value = 800; // Threshold CO2 gas sensor value (adjust as needed)


void setup() {
  Serial.begin(9600); // Initialize serial communication
  Wire.begin();      // Initialize the Wire library for I2C communication

  rtc.begin();       // Start RTC module using the default Wire library


  pinMode(triggerPin, OUTPUT); // Set trigger pin as output
  pinMode(echoPin, INPUT);    // Set echo pin as input
```

```

scale.begin(weightDataPin, weightClockPin); // Initialize weight sensor
scale.set_scale(); // Calibrate the weight sensor
scale.tare(); // Reset the scale to 0
}

void loop() {
    long duration, distance;
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = duration * 0.034 / 2;

    DateTime now = rtc.now();

    float weight = scale.get_units(); // Read weight sensor data in grams
    int methaneValue = analogRead(methaneSensorPin); // Read methane gas sensor data
    int co2Value = analogRead(co2SensorPin); // Read CO2 gas sensor data

    if (distance <= thresholdDistance && weight >= thresholdWeight &&
        methaneValue >= thresholdMethaneValue && co2Value >= thresholdCO2Value &&
        now.hour() == 12) {
        // Your logic here for garbage collection
    }

    if (now.second() % 2 == 0) {
        // Simulate some action if needed
    }
}

```

```
Serial.print("Distance: ");  
Serial.print(distance);  
Serial.print(" cm | Weight: ");  
Serial.print(weight);  
Serial.print(" g | Methane Value: ");  
Serial.print(methaneValue);  
Serial.print(" | CO2 Value: ");  
Serial.print(co2Value);  
Serial.print(" | Time: ");  
Serial.println(now.timestamp());  
  
delay(1000); // Delay for stability  
}
```