

Listing Program

A. Master

```
#include <Servo.h>
#define PIR_PIN 2
#define TOUCH_SENSOR_PIN 3
#define VIBRATION_SENSOR_PIN 4
#define SERVO_PIN 9
Servo myservo;
void setup() {
  Serial.begin(9600);
  pinMode(PIR_PIN, INPUT);
  pinMode(TOUCH_SENSOR_PIN, INPUT);
  pinMode(VIBRATION_SENSOR_PIN, INPUT);
  myservo.attach(SERVO_PIN);
  myservo.write(0); // Servo in closed position
}
void loop() {
  if (digitalRead(TOUCH_SENSOR_PIN) == HIGH) {
    // System off
    myservo.write(0);
    delay(500);
    while (digitalRead(TOUCH_SENSOR_PIN) == HIGH); // Wait until
    touch sensor is released
  } else {
    handlePIR();
    handleVibration();
  }
}
void handlePIR() {
  if (digitalRead(PIR_PIN) == HIGH) {
    myservo.write(90); // Open the box
    delay(5000); // Keep the box open for 5 seconds
    myservo.write(0); // Close the box
  }
}
void handleVibration() {
  int vibrationValue = analogRead(VIBRATION_SENSOR_PIN);
  if (vibrationValue > 500) {
    Serial.println("ALARM"); // Send alarm signal to Slave
    delay(1000);
  }
}
```

B. Slave

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>
#define TRIGGER_PIN 2
#define ECHO_PIN 3
#define RFID_SS_PIN 8
```

```

#define RFID_RST_PIN 9
#define BUZZER_PIN 4
#define SOLENOID_PIN 5
LiquidCrystal_I2C lcd(0x27, 16, 2);
MFRC522 rfid(RFID_SS_PIN, RFID_RST_PIN);
Servo myservo;
void setup() {
  Serial.begin(9600);
  pinMode(TRIGGER_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  lcd.begin();
  lcd.backlight();
  SPI.begin();
  rfid.PCD_Init();
  myservo.attach(SERVO_PIN);
  myservo.write(0); // Servo in closed position
}
void loop() {
  if (Serial.available()) {
    String message = Serial.readString();
    if (message == "ALARM") {
      digitalWrite(BUZZER_PIN, HIGH);
      delay(1000);
      digitalWrite(BUZZER_PIN, LOW);
    }
  }
  handleUltrasonic();
  handleRFID();
}
void handleUltrasonic() {
  long duration, distance;
  digitalWrite(TRIGGER_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIGGER_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGGER_PIN, LOW);
  duration = pulseIn(ECHO_PIN, HIGH);
  distance = (duration / 2) / 29.1;
  if (distance < 5) {
    lcd.clear();
    lcd.print("Box is Full");
  } else {
    lcd.clear();
    lcd.print("Donate Please");
  }
}
void handleRFID() {
  if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) {
    return;
  }
}

```

```
}  
// Assuming we have a valid card ID stored  
String validID = "YOUR_VALID_ID_HERE";  
String readID = "";  
for (byte i = 0; i < rfid.uid.size; i++) {  
  readID += String(rfid.uid.uidByte[i] < 0x10 ? " 0" : " ");  
  readID += String(rfid.uid.uidByte[i], HEX);  
}  
if (readID == validID) {  
  myservo.write(90); // Open the box  
  delay(5000); // Keep the box open for 5 seconds  
  myservo.write(0); // Close the box  
}  
rfid.PICC_HaltA();  
rfid.PCD_StopCrypto1();  
}
```