Smart Intercom System - Arduino Code

```
#include <LiquidCrystal.h>
#include <Servo.h>

#define SERVO_PIN 12
#define RGB_RED_PIN 11
#define RGB_GREEN_PIN 10
#define RGB_BLUE_PIN 9
#define BUZZER_PIN 8
#define BUTTON_PIN 7
#define LCD_RS_PIN A4
#define LCD_E_PIN A5
#define LCD_D4_PIN 2
#define LCD_D5_PIN 3
#define LCD_D6_PIN 4
#define LCD_D7_PIN 5

#define SERVO_OPEN_DOOR_POSITION 50
#define SERVO_CLOSE_DOOR_POSITION 140

LiquidCrystal lcd(LCD_RS_PIN, LCD_E_PIN, LCD_D4_PIN,
                            LCD_D5_PIN, LCD_D6_PIN, LCD_D7_PIN);
Servo servo;

unsigned long lastTimeButtonChanged = millis();
unsigned long debounceDelay = 200;  // Increased debounce delay for better filtering
byte previousButtonState;
bool buttonPressed = false;

void setup() {
  Serial.begin(115200);
  servo.attach(SERVO_PIN);
  servo.write(SERVO_CLOSE_DOOR_POSITION);

  pinMode(BUTTON_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(RGB_RED_PIN, OUTPUT);
  pinMode(RGB_GREEN_PIN, OUTPUT);
  pinMode(RGB_BLUE_PIN, OUTPUT);

  previousButtonState = digitalRead(BUTTON_PIN);

  lcd.begin(16,2);
  lcd.print("Starting...");
  delay(1000);
  lcd.clear();
  lcd.print("Push button to call");
}

void loop() {
  unsigned long timeNow = millis();
  if (timeNow - lastTimeButtonChanged >= debounceDelay) {
    byte buttonState = digitalRead(BUTTON_PIN);
    if (buttonState != previousButtonState) {
      lastTimeButtonChanged = timeNow;
      previousButtonState = buttonState;

      if (buttonState == HIGH && !buttonPressed) {
        buttonPressed = true;
        Serial.println("button_pressed");
      }
    }
  }

  if (Serial.available() > 0) {
```

```
    String cmd = Serial.readStringUntil('\n');
    if (cmd == "open_door") {
      servo.write(SERVO_OPEN_DOOR_POSITION);
    }
    else if (cmd == "close_door") {
      servo.write(SERVO_CLOSE_DOOR_POSITION);
    }
    else if (cmd.startsWith("print_text:")) {
      cmd.remove(0, 11);
      String line1 = cmd.substring(0, 16);
      String line2 = cmd.substring(16);
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print(line1);
      lcd.setCursor(0, 1);
      lcd.print(line2);
    }
    else if (cmd.startsWith("play_buzzer:")) {
      cmd.remove(0, 12);
      int comaIndex = cmd.indexOf(',');
      int frequency = cmd.substring(0, comaIndex).toInt();
      int duration = cmd.substring(comaIndex + 1).toInt();
      tone(BUZZER_PIN, frequency, duration);
    }
    else if (cmd.startsWith("set_led:")) {
      cmd.remove(0, 8);
      int red = cmd.substring(0, 3).toInt();
      int green = cmd.substring(4, 7).toInt();
      int blue = cmd.substring(8).toInt();
      analogWrite(RGB_RED_PIN, red);
      analogWrite(RGB_GREEN_PIN, green);
      analogWrite(RGB_BLUE_PIN, blue);
    }
  }
}
```