**Short Introduction to Arduino Code Used Here:**

To understand the provided Arduino code, it is essential to grasp some basics of Arduino coding, especially the data types used.

**Data Types:**

The data types used in this project are const int, int, long, and Servo

**const int:**

* Description: const int is used to declare constants. These are integer values that do not change during the execution of the program. The const keyword ensures that the value of these variables cannot be altered.
* Usage: In this code, const int is used to define the pin numbers for various components such as the gas sensor, LED, ultrasonic sensor, servo motor, light sensor, and LEDs.

**Example:**

const int gasSensorPin = A0;

const int gasLEDPin = 12;

**int:**

* Description: int is a data type used to store integer values. Integers are whole numbers without a fractional component.
* Usage:
* ◦ GAS\_VAL stores the value read from the gas sensor.
* ◦ distance stores the calculated distance from the ultrasonic sensor.
* ◦ angle sets the initial angle of the servo motor.
* ◦ analogValue stores the analog value read from the light sensor.

**Example:**

int GAS\_VAL = 0;

int distance;

**long:**

* Description: long is a data type used to store large integer values. It can hold a larger range of numbers than int.
* Usage: duration stores the time it takes for the ultrasonic sensor's pulse to travel to an object and back, measured in microseconds.

**Example:**

long duration;

**Servo:**

* Description: Servo is a data type defined by the Servo library. It is used to create a servo object that can control a servo motor.
* Usage: servo is an instance of the Servo class, which provides functions to control a servo motor's position.

**Example:**

Servo servo;

**setup() Function:** The setup() function is a crucial part of an Arduino sketch, executed only once when the microcontroller (like Arduino) starts up.

**Purpose of setup() Function:**

* Initialization: Prepares the environment for the main execution loop (loop() function) of the Arduino program.
* Configuration: Configures pins, initializes communication interfaces, and sets initial states for components such as servo motors and LEDs.
* Feedback: Provides feedback via the Serial Monitor to indicate initialization status and ensures readiness before the main program logic in loop() begins execution.

Description of setup() Function:

1. Serial Communication Initialization (Serial.begin(9600)):

* Initializes a serial communication channel at 9600 bits per second between Arduino and a computer. This allows data transmission for debugging and monitoring.
* The baud rate of 9600 is widely supported and balances speed with reliability.

1. Pin Mode Initialization:

* pinMode(pinNumber, INPUT): Configures pinNumber as an input.
* pinMode(pinNumber, OUTPUT): Configures pinNumber as an output.
* Example: pinMode(7, INPUT); pinMode(10, OUTPUT);

1. Servo Motor Initialization:

* servo.attach(servoPin): Initializes the servo motor connected to servoPin.
* servo.write(angle): Sets the initial position of the servo motor to angle.

1. Print Setup Complete Message:

* Serial.println("Setup complete"); Sends "Setup complete" to the Serial Monitor, indicating successful execution of all setup() function steps.

A concise reference for the key functions and statements used in the loop() function of this projects code.

**Reading Analog Sensor Value:**

file.cpp

analogValue = analogRead(pinNumber);

* **Syntax:** analogRead(pinNumber)
* **Description:** Reads the analog value from the specified pin and stores it in a variable.

**Serial Communication:**

file.cpp

Serial.print("Message");

Serial.println(variable);

* **Syntax:** `Serial.print(value)`
* **Description:** Sends data to the Serial Monitor without a newline.
* **Syntax:** `Serial.println(value)`
* **Description:** Sends data to the Serial Monitor with a newline.

**Conditional Statement:**

File.cpp

if (condition) {

// statements

} else {

// statements

}

* **Syntax:** `if (condition) { // statements } else { // statements }`
* Description: Executes different code blocks based on the evaluation of a condition.

**Setting Digital Output:**

File.cpp

digitalWrite(pin, value);

* **Syntax:** `digitalWrite(pin, value)`
* **Description:** Sets the digital output of a specified pin to HIGH or LOW.

**Microsecond Delay:**

file.cpp

delayMicroseconds(us);

* **Syntax:** `delayMicroseconds(us)`
* **Description:** Pauses the program for a specified number of microseconds.

**Pulse Duration Measurement:**

file.cpp

duration = pulseIn(pin, value);

* **Syntax:** `pulseIn(pin, value)`
* **Description:** Measures the duration of a pulse on the specified pin and returns the time in microseconds.

Servo Control:

File.cpp

servo.write(angle);

* **Syntax:** `servo.write(angle)`
* **Description:** Sets the servo position to a specified angle.

**Millisecond Delay:**

File.cpp

delay(ms);

* **Syntax:** `delay(ms)`
* **Description:** Pauses the program for a specified number of milliseconds.

These syntax descriptions provide a concise reference for the key functions and statements used in the provided `loop()` function.