

## Components Used and Their Functions

### **1. Arduino UNO**

#### **Function:**

Acts as the main microcontroller that reads sensor data and sends it to the computer.

#### **Technical Role:**

1. Controls the ultrasonic sensors.
2. Generates trigger pulses and measures echo duration.
3. Converts echo time to distance in centimeters.
4. Sends continuous distance data to the laptop over USB serial communication.
5. Provides stable timing and processing needed for real-time gesture detection.

### **2. Ultrasonic Sensors (HC-SR04) - 2 Units**

Used as the primary gesture detection sensors.

#### **Function:**

Measure the distance between the user's hand and the laptop/sensor.

#### **Technical Role:**

1. Each sensor:
  - a. Sends ultrasonic waves at 40 kHz.
  - b. Waves hit the hand and bounce back.
2. Sensor calculates distance using time-of-flight:
  - a. Distance = time x 0.034 / 2
3. Two sensors provide two-dimensional gesture control:
  - a. Right Sensor: Scroll + Next Tab.
  - b. Left Sensor: Previous Tab + Play/Pause.
  - c. Both together: Task Switching.
4. The dual-sensor setup increases accuracy and enables multiple gesture types.

### **3. Jumper Wires**

#### **Function:**

Provide electrical connections between Arduino and ultrasonic sensors.

#### **Technical Role:**

1. 4 wires per sensor (VCC, GND, Trig, Echo)
2. Maintain stable connections for accurate signal timing
3. Transfer high-frequency pulses without distortion
4. Even minor loose connections can cause false gesture detection, so they are critical.

### **4. USB Cable (Type B)**

#### **Function:**

Connects Arduino to the laptop.

#### **Technical Role:**

1. Transfers serial data from Arduino to Python.
2. Provides power (5V) to run Arduino.
3. Ensures continuous real-time data streaming for gesture recognition.

## **5. Laptop or PC**

### **Function:**

Acts as the control unit where Python software runs and performs actions.

### **Technical Role:**

1. Executes the Python script that processes sensor data.
2. Uses automation library to trigger keyboard/mouse events.
3. Host for the Arduino serial port.
4. Runs media players or browsers being controlled.

## **6. Python Programming Language**

### **Function:**

Processes incoming sensor data and converts gestures into actions.

### **Technical Role:**

1. Reads serial data from Arduino.
2. Applies gesture detection rules.
3. Determines if user is scrolling, switching tabs, or performing play/pause.
4. Sends software-level events to the OS.
5. Python acts as the brain that interprets gestures logically.

## **7. PyAutoGUI Library (Python)**

### **Function:**

Simulates keyboard and mouse actions.

### **Technical Role:**

1. Performs scrolling:
  - a. `pyautogui.scroll()`
2. Switches browser tabs:
  - a. `pyautogui.hotkey("ctrl", "tab")`
3. Plays/pauses videos:
  - a. `pyautogui.press("space")`
4. Performs task switching (Alt + Tab).
5. It bridges the gap between gesture detection and actual computer control.

## **8. PySerial Library (Python)**

### **Function:**

Connects Python to Arduino using serial communication.

### **Technical Role:**

1. Opens the COM port.
2. Reads incoming distance data (e.g., "23,41")
3. Converts raw serial data into integers.
4. Ensures smooth, real-time communication between hardware and software.
5. Without PySerial, gesture data would not reach Python.

## **9. Arduino IDE**

### **Function:**

Used to write, upload, and debug the Arduino code.

### **Technical Role:**

1. Compiles the C++ code for the ATmega328P microcontroller.
2. Uploads gesture distance measurement program.
3. Provides Serial Monitor for debugging sensor values.

## **10. Laptop Mount / Sensor Holder (optional)**

1. Keeps ultrasonic sensors fixed in stable position.
2. Ensures consistent gesture detection.

## **11. Breadboard (optional)**

1. Helps in neat wiring.
2. Adds stability in connections.