# Smart Toilet System: An Arduino-Based Project Report

**Author: Rakesh Pittala, Taranisen Naik, Aditya Pradhan**

Date: November 13, 2025

## 1. Abstract

This report details the design, implementation, and testing of a prototype "Smart Flush System" built using an Arduino microcontroller. The primary objective is to develop an automated, touchless system that intelligently prepares a flush mechanism. The system employs two HC-SR04 ultrasonic sensors and one SG90 servo motor. The first ultrasonic sensor is positioned to detect the presence of a user sitting on the toilet. The second ultrasonic sensor is mounted in the flush tank to monitor the water level. A servo motor is configured to activate—rotating to compress an external spring—only when two specific conditions are met simultaneously: 1) a user is detected in the "sitting" position, and 2) the flush tank's water level is "full." This logic ensures the system only "arms" itself when a flush is possible and necessary, and remains inactive if the water level is low (e.g., after a manual flush). This document covers the hardware selection, circuit design, software logic, and operational procedure of the prototype.

## 2. Introduction

### 2.1 Problem Statement

Traditional manual-flush toilets present two significant challenges: hygiene and water conservation. The flush handle is a high-contact surface, contributing to the spread of germs and bacteria. Furthermore, users may flush with an empty or partially filled tank, leading to an ineffective flush and wasted water, or may forget to flush entirely.

### 2.2 Project Objective

The objective of this project is to design and build a cost-effective, automated smart toilet system that addresses these issues. The core goals are:

* **Enhance Hygiene:** To create a touchless system that prepares for a flush without user intervention.
* **Conserve Water:** To build "smart logic" into the system that prevents activation unless the flush tank is almost full, ensuring an effective flush and preventing wasted water from "half-flushes."

## 3. System Design and Components

The system is composed of two main parts: the hardware components and the software logic that governs their interaction.

### 3.1 Hardware Components

* **Microcontroller (1x Arduino Uno):** The "brain" of the project. It processes input from both sensors and sends the output signal to the servo motor.
* **User Detection Sensor (1x HC-SR04 Ultrasonic Sensor):** Positioned to measure the distance in front of it. It detects if an object (the user) is within a specific range that corresponds to a "sitting" position.
* **Water Level Sensor (1x HC-SR04 Ultrasonic Sensor):** Mounted at the top of the flush tank, pointing downwards. It measures the distance to the water's surface. A small distance reading corresponds to a "full" tank.
* **Actuator (1x SG90 Servo Motor):** A small motor used for precise rotational control. In this project, it rotates to a specific angle (90° in our case) to engage the spring mechanism and returns to a resting angle (0°) when conditions are not met.
* **Miscellaneous:** Jumper wires, breadboard.

### 3.2 System Architecture (manually)

The operational flow is as follows:

1. The Arduino continuously polls both ultrasonic sensors for distance readings.
2. **Sensor 1 (User):** The distance is compared to a **personThreshold**. If the (base distance-person distance) is greater than this threshold, a boolean variable isPersonSitting is set to true.
3. **Sensor 2 (Water):** The distance is compared to a **flushThreshold**. If new water distance -initial distance is greater than this threshold, then flush is already done.
4. **Control Logic:** Once the boolean is set to true (person presence is found), the flush distance is measured. After the person has left (boolean false), there is a 5s delay, and if the water level distance-based distance is less than the threshold, then flush occurs.

## 4. Implementation(Manually)

### 4.1 Circuit Assembly

The components are connected as follows:

* **Power:** All component VCC pins are connected to the Arduino's 5V pin. All GND pins are connected to the Arduino's GND pin.
* **User Sensor (Sensor 1):**
  + Trig pin is connected to Arduino Digital Pin 2.
  + Echo pin is connected to Arduino Digital Pin 3.
* **Water Sensor (Sensor 2):**
  + Trig pin is connected to Arduino Digital Pin 4.
  + Echo pin is connected to Arduino Digital Pin 5.
* **Servo Motor:**
  + Signal pin (usually orange or yellow) connected to Arduino PWM Pin 9.

## 5. Testing and Results

### 5.1 Calibration

Calibration is the most critical step. The Arduino's Serial Monitor was used to determine the correct threshold values.

* **Person Sensor:** The distance was measured while a user was sitting. This "sitting" distance was found to be approximately 8 cm. The threshold was set to 10 cm to provide a reliable buffer.
* **Water Sensor:** The sensor was mounted in the tank. The distance to the water surface when "full" was 5 cm. The threshold was set to 6 cm. When the tank was flushed, this distance increased to over 20 cm.

### 5.2 Scenario Testing

The system was tested against all possible operational scenarios:

| **Scenario** | **Person Sitting? (distance)** | **Water Level** | **Expected Servo Position** | **Observed Result** |
| --- | --- | --- | --- | --- |
| 1. Idle | 18cm | 5cm | (0°) | **Pass** |
| 2. User Sits | 8cm | 5cm | (90°) | **Pass** |
| 3. Manual Flush (User Sitting) | 9cm | 11cm | (0°) | **Pass** |

### 5.3 Results

The system performed exactly as designed. The servo motor reliably moved to the 90-degree "compress" position **only** when a person was sitting *and* the water tank was full. In all other states—including when the person was sitting but the tank was refilling (after a manual flush)—the servo correctly returned to or remained at the 0-degree "release" position.

### 6. Conclusion and Future Scope

While the electronic prototype is successful, further development could include:

* **Mechanical Linkage:** Designing and 3D-printing a durable, custom-fit mechanical linkage to connect the servo motor to the spring and flush lever.
* **Waterproofing:** Creating a sealed, waterproof enclosure for the Arduino and sensors to ensure long-term reliability in a humid bathroom environment.
* **Logic Inversion (Flush on Exit):** A popular alternative logic would be to detect when the isPersonSitting state changes from true to false. At that "exit event," the system would check if (isWaterFull) and *then* trigger the servo to perform a single, complete flush action (e.g., move 0 -> 90 -> 0) rather than just holding a position.
* **Manual Override:** Adding a physical push-button as a manual override, allowing a user to trigger the servo even if the sensors fail or are bypassed.

**Code Implementation**- <https://github.com/TaranisenNaik/smart-flush>