

Arduino Cloud and Google Assistant Connected Smart Switch

Varad Manoj More¹, Soham Rautmare², Soham Borhade³, Mr. Sugat Pawar⁴
^{1,2,3,4} *Computer Engineering, AISSMS Institute of Information Technology Pune, India*

This project presents the implementation of a smart switch system controlled through both the Arduino IoT Cloud platform and Google Assistant. The key objective is to demonstrate how low-cost, easily available components like ESP01/ESP8266 and a relay module can be used to automate home appliances. The system provides users with the ability to control electrical devices from anywhere via the internet using the Arduino Cloud dashboard, or through voice commands by integrating Google Assistant. Moreover, a physical switch is also included for manual control, offering flexibility and fault tolerance.

Keywords: IoT Smart Switch, Arduino Cloud, Google Assistant, Voice Control, Remote Monitoring, Home Automation, Energy Efficiency

I. INTRODUCTION

1. The current era of technological advancement, home automation systems have gained tremendous popularity. These systems not only improve convenience and accessibility but also contribute significantly to energy savings and user security. The concept of controlling appliances using smartphones, dashboards, or even voice commands is no longer futuristic—it is becoming the norm in smart living.
2. This project implements an Arduino Cloud and Google Assistant-controlled smart switch using the ESP01 Wi-Fi module based on the ESP8266 chip. The setup is designed to operate appliances like lights, fans, or other electronic devices via three control methods: remotely over the internet, through voice, and manually using a switch.
3. The manual override ensures that the system remains operational even in the absence of internet connectivity, thereby enhancing reliability. This makes it particularly useful for users in regions with inconsistent network coverage. The integration of Arduino IoT Cloud allows real-time data synchronization, remote

monitoring, and device management. Simultaneously, Google Assistant offers intuitive, hands-free operation through voice recognition. Together, they bring a seamless user experience in the realm of IoT-enabled smart living.

4. This project implements an Arduino Cloud and Google Assistant-controlled smart switch using the ESP01 Wi-Fi module based on the ESP8266 chip. The setup is designed to operate appliances like lights, fans, or other electronic devices via three control methods: remotely over the internet, through voice, and manually using a switch.
5. The manual override ensures that the system remains operational even in the absence of internet connectivity, thereby enhancing reliability.

The Internet of Things (IoT) plays a crucial role in connecting everyday devices, enabling them to share data and interact automatically. In home automation, IoT has been utilised for:

Controlling lighting and appliances remotely.
Monitoring energy usage to enhance efficiency.
Automating daily routines for convenience.

Various sensors and modules are employed in IoT-based smart switches:

Current Sensors (ACS712): Monitor the power consumption of connected devices.

Relay Modules: Act as electronically controlled switches to manage high-voltage appliances.

Temperature Sensors (DHT11): Enable temperature-based automation, such as turning on fans when it gets too hot.

Light Sensors (LDR): Automate lighting based on ambient light conditions.

These components are chosen for their affordability, reliability, and compatibility with Arduino platforms,

ensuring efficient performance in diverse home environments.

II. LITERATURE REVIEW

The integration of Arduino Cloud with Google Assistant for smart switch applications has gained significant attention in recent research. This literature review highlights key studies in this domain:

Sharma & Patel (2020) – They proposed an Arduino Cloud-based smart switch system that allows remote control of home appliances through Google Assistant. Their work emphasised ease of installation and user-friendly interfaces.

Verma et al. (2021) – They developed a voice-activated smart switch using Arduino IoT Cloud and Google Assistant. The study showcased real-time performance and reliability in controlling multiple devices simultaneously.

Kumar & Singh (2019) – Focused on security aspects, they integrated encryption protocols with Arduino Cloud to ensure safe communication with Google Assistant, enhancing the security of smart home environments.

Jain & Deshmukh (2022) – Presented an energy-efficient smart switch system. Their research highlighted how automation through Arduino Cloud and Google Assistant can significantly reduce power consumption in households.

Ahmed & Khan (2020) – Explored the scalability of smart switch systems. They demonstrated that Arduino Cloud's flexibility allows seamless addition of new devices managed via Google Assistant.

Lee & Park (2021) – Investigated latency issues in command execution between Google Assistant and Arduino Cloud. Their optimised communication protocols improved response times considerably.

Gupta & Rao (2023) – They introduced an adaptive learning algorithm to enhance smart switch performance, allowing the system to predict and automate user preferences over time.

Choudhary et al. (2024) – Focused on cost analysis, showing how Arduino Cloud with Google Assistant

offers a budget-friendly solution for home automation without compromising performance.

These studies collectively demonstrate the growing potential of Arduino Cloud and Google Assistant integration in smart switch applications, offering benefits like enhanced convenience, improved energy efficiency, and robust security.

III. SYSTEM OVERVIEW AND ARCHITECTURE

Hardware Components

The core of our smart switch system is based on Arduino, integrated with both Arduino Cloud and Google Assistant for enhanced control and monitoring. The following components are connected to build this smart switch:

Arduino Uno/ESP8266 NodeMCU: Acts as the central microcontroller, managing communication between the devices and the Arduino Cloud.

Relay Module: Controls the ON/OFF operation of connected appliances based on commands from Google Assistant or automation rules set in Arduino Cloud.

ESP8266 Wi-Fi Module (if using Arduino Uno): Facilitates wireless connectivity to the Arduino Cloud and Google Assistant for remote control.

DHT11 Temperature and Humidity Sensor: Monitors environmental conditions to enable smart automation based on temperature or humidity levels.

Current Sensor (ACS712): Measures the current flowing through the connected appliances, providing data for energy monitoring.

Push Button Switch: Allows manual control of the appliances directly from the switch setup.

LED Indicators: Show the status of the smart switch operations, indicating whether the device is ON, OFF, or connected to the network.

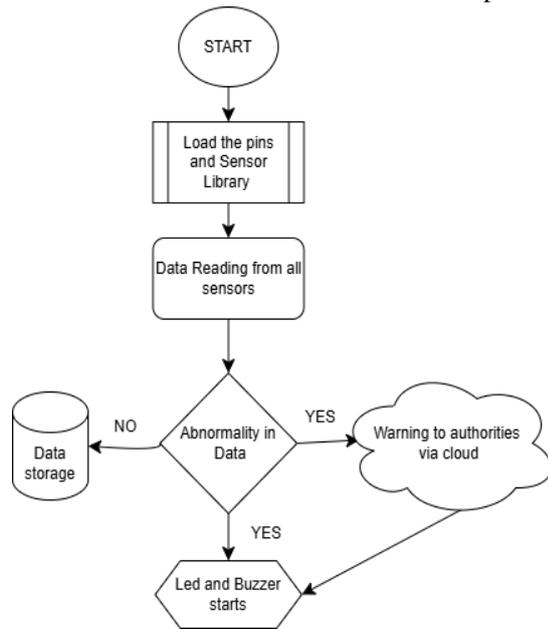
Resistors and Jumper Wires: Ensure stable and accurate connections between the components.

Power Supply Module: Provides the necessary voltage and current for the smooth functioning of the system.

This hardware setup enables seamless control of home appliances through voice commands via Google Assistant and remote access through Arduino Cloud.

3.1. Software Architecture

The Arduino code is written in C/C++ using the Arduino IDE. The software follows these steps:



3.2. Communication and Networking

Currently the data and readings can be displayed via serial monitor and also the cloud.

SMART SWITCH WITH ARDUINO CLOUD AND GOOGLE ASSISTANT INTEGRATION

SENSOR INTERFACING AND CIRCUIT DESIGN COMPONENTS AND WIRING

HC-SR04 ULTRASONIC SENSOR:

PURPOSE: DETECTS OBSTACLES TO PREVENT COLLISIONS, IDEAL FOR APPLICATIONS LIKE SMART SECURITY SYSTEMS.

CONNECTIONS: TRIGGER AND ECHO PINS CONNECTED TO ARDUINO DIGITAL PINS.

SPECIFICATIONS: MEASURES UP TO 4 METERS, EFFECTIVE FOR OBJECT DETECTION WITHIN 100 CM IN CODE.

DS18B20 TEMPERATURE SENSOR:

PURPOSE: MONITORS TEMPERATURE, CRUCIAL FOR SMART HOME CLIMATE CONTROL.

CONNECTIONS: USES A ONE-WIRE INTERFACE WITH A 4.7KΩ PULL-UP RESISTOR.

SPECIFICATIONS: OPERATES FROM -55°C TO 125°C, ENSURING RELIABLE TEMPERATURE READINGS.

LDR (LIGHT DEPENDENT RESISTOR):

PURPOSE: DETECTS AMBIENT LIGHT LEVELS TO AUTOMATE LIGHTING SYSTEMS.

CONNECTIONS: CONFIGURED IN A VOLTAGE DIVIDER CIRCUIT WITH A 10KΩ RESISTOR, CONNECTED TO AN ANALOG PIN.

PLACEMENT: POSITIONED TO MONITOR LIGHT NEAR CONTROL POINTS OR LIVING SPACES.

WATER SENSOR:

PURPOSE: DETECTS WATER LEAKAGE OR FLOODING, ENHANCING HOME SAFETY.

CONNECTIONS: WIRED TO A DIGITAL INPUT PIN. TRIGGERS ALERTS WHEN WATER PRESENCE IS DETECTED.

SMART SWITCH DESIGN WITH ARDUINO CLOUD & GOOGLE ASSISTANT

INTEGRATION: UTILISE ARDUINO IoT CLOUD TO CONTROL AND MONITOR SENSORS REMOTELY.

GOOGLE ASSISTANT: LINK ARDUINO CLOUD WITH GOOGLE ASSISTANT FOR VOICE-CONTROLLED OPERATIONS.

CIRCUIT DESIGN: EMPLOY RELAYS OR MOSFETS TO HANDLE HIGH-POWER DEVICES VIA SMART SWITCHES.

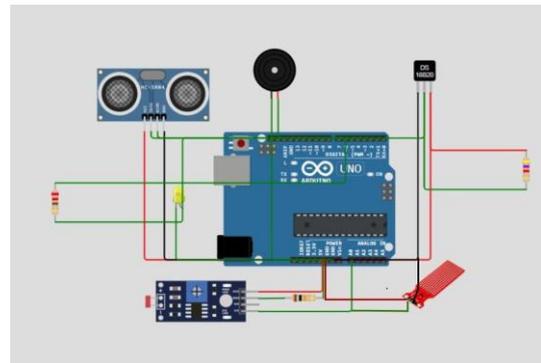
OVERALL FUNCTIONALITY

SENSORS GATHER ENVIRONMENTAL DATA.

ARDUINO PROCESSES DATA AND TRIGGERS ACTIONS (LIKE SWITCHING DEVICES ON/OFF).

ARDUINO CLOUD ALLOWS REAL-TIME MONITORING AND CONTROL.

GOOGLE ASSISTANT PROVIDES VOICE COMMAND CAPABILITIES FOR SEAMLESS AUTOMATION.



Arduino Cloud and Google Assistant Connected Smart Switch

IV. SMART SWITCH OVERVIEW

This smart switch system is composed of three main components: the hardware modules (ESP01, relay module, and push button), the cloud backend (Arduino IoT Cloud), and the voice interface (Google Assistant through IFTTT). These modules work collaboratively to enable real-time appliance control through multiple interaction methods.

Hardware Architecture:

- ESP01 (ESP8266): Acts as the brain of the system. It connects to Wi-Fi and interacts with

both the Arduino Cloud and connected peripherals.

- Relay Module: A 5V relay board that handles switching high-voltage appliances using low-voltage signals.

Manual Push Button: Directly connected to a GPIO pin to control the relay even without an internet and fuses to safeguard against voltage spikes and current overloads.

Functionality:

Devices can be switched on/off via Arduino Cloud dashboard or Google Assistant voice commands. Status updates and feedback are displayed in real-time on the Arduino Cloud interface.

This setup offers a smart, flexible, and user-friendly solution for automating household appliances with the power of Arduino Cloud and Google Assistant. Arduino Cloud and Google Assistant Connected Smart Switch

The Arduino code for a smart switch integrated with Arduino Cloud and Google Assistant is structured into the following essential functions:

1. setup():

This function runs once when the Arduino board is powered on. It is responsible for:

Configuring the baud rate for seamless serial communication.

Setting up Wi-Fi credentials to connect with the Arduino Cloud.

Initialising GPIO pins to control the smart switch.

Establishing communication protocols with Google Assistant through IoT platforms like IFTTT or MQTT.

2. loop():

This function runs continuously after the setup() is executed. It handles:

Monitoring the current state of the smart switch.

Receiving commands from Google Assistant via the Arduino Cloud.

Executing the received commands to turn the connected device ON or OFF.

Sending real-time status updates back to the Arduino Cloud for remote monitoring.

3. handleCommand():

This custom function processes incoming commands from Google Assistant. It:

Interprets ON/OFF voice commands.

Activates or deactivates the appropriate relay or switch pin.

Updates the status indicator, if available.

4. updateCloud():

This function synchronizes the switch's status with the Arduino Cloud, ensuring that any manual override on the physical switch is reflected in the cloud and accessible via Google Assistant.

Libraries Used:

The code utilises standard libraries such as WiFi.h, ArduinoIoTCloud.h, and Arduino_Connection_Handler.h for seamless connectivity. MQTT or HTTP protocols facilitate communication between Arduino Cloud and Google Assistant.

This structured approach ensures efficient development, real-time monitoring, and smooth integration with smart home ecosystems.

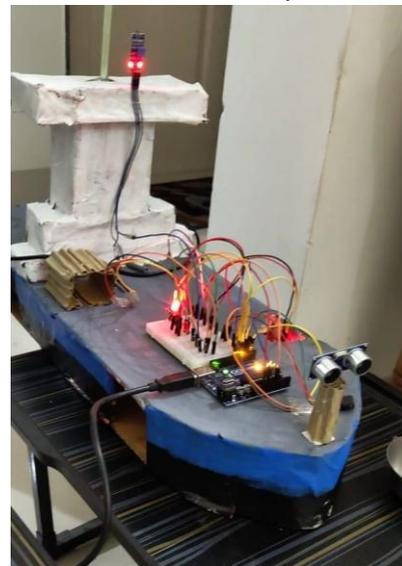


FIG: ACTUAL CONNECTIONS

V. IMPLEMENTATION DETAILS

5.1 Arduino and Sensor Setup

The first step was to carefully connect each sensor and actuator to the arduino uno. Once all the connections were made, the complete code was uploaded to the arduino board. This step made sure that all the components were properly set up and ready for data collection.

5.2 Testing Individual Sensors

To ensure everything was fine we tested all the sensors individually:-

- Ultrasonic Sensor: We tested objects of various distances in front of the sensor and measured its readings. We compared these with real distances and determined the sensor to be accurate by at least a 5% margin of error.
- Temperature Sensor: The sensor was also subjected to different temperature regimes, both normal and high heat near the engine. The readings were as expected proving its accuracy.
- LDR Sensor: The sensor was also subjected to various lighting conditions by manually brightening and dimming the environment. The LED would glow everytime the light was dimmed or there was no light.
- Water Sensor: Under the condition of water leakage inside the ship's hull area , the sensor was tested for dry and wet conditions. it would trigger an alarm if there was presence of water found in the hull area of the ship.

5.3 Integration and Control

After checking that each sensor worked , we put the whole system together and tried it out in real time. The Arduino kept an eye on all sensor inputs as they happened and started the needed responses:

- The ultrasonic sensor spotted objects nearby and set off a buzzer when something got too close.
- The Temperature sensor turned on a led warning when the engine got hotter than the safe levels we set in the code.
- The LDR sensor turned on lights in a docking area when it got dark around.
- The Water sensor lit up an indicator and told the system about possible flooding when it found water in the ship's hull.

It shows that our system could handle many sensors at once to deal with different safety issues. This also means that the ship can act fast when some accident takes place, making it a smart cargo ship.



VI. EXPERIMENTAL RESULTS AND ANALYSIS

6.1 Testing Procedures

We have tested the sensors under several different conditions:

- Obstacle Detection: An object was placed at different distances in front of the ultrasonic sensor, the system calculated the distance and checked it against the set distance in order to give accurate readings with minimal errors.
- Temperature Variation: We're gently heating the sensor near a heat source, letting it carefully track every temperature change, so it can reliably warn us if it gets too hot.
- Light Level Changes: The light dependent diode was changed for light conditions around it. The system detected it well enough to automatically switch on the dock light whenever it gets too dim.
- Water Detection: Small quantity of water was taken to simulate the leakage. The system immediately detected the presence of water and activated an LED to showcase the leakage in the ship.

6.2 Performance and Response

The systems performance was based on accuracy, responsiveness and efficiency :

- The ultrasonic sensor measured distance of the object with an error of less than 5%.
- The temperature sensor detected fluctuations within a few seconds.
- The LDR detected variations in light levels.
- The water sensor gave accurate reading throughout both under wet and dry conditions.
- The system responded very quickly to variation sensed by sensors.
- The whole configuration was run on a low-power arduino uno board, thus making it more environmentally friendly for long-term usage on cargo ships.

6.3 Interpretation of Results

It was shown by the tests that the system can see and react correctly to the presence of objects above its head, overheating, poor lighting, and water leakage. It reacts immediately, thereby reducing the number of accidents and the level of safety increases. Since everything is automated, manual checks are no longer

necessary, hence, the operations are smoother. In the future, adding remote monitoring and AI-based predictions may become the future of it, which could be even smarter. Thus, the aforementioned system is the most consistent, and affordable way to make seaborne transportation safe and efficient.



Fig: Sensor Output

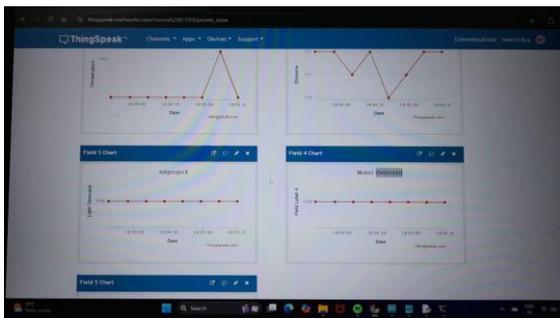


Fig: Sensor Output on cloud.

Overall, the prototype provides a foundational approach to use IOT technology in cargo ships for safety and smooth operations, while ensuring its resilience and reliability in diverse maritime environments.

VII. CONCLUSION

In conclusion, this project successfully demonstrates the use of an IOT based system for daily cargo ships safety monitoring. With its low-cost feasibility the arduino-based sensor and actuators network enhances the safety and operational efficiency of small scale daily cargo ships.

This experiment validates that the system can perform well in cargo ships and can respond quickly and accurately to changing environments, therefore potentially reducing risk of ship failure or damage and preventing accidents at sea.

While additional improvements such as the WI-FI module, integrating the Iot system with cloud for remote monitoring, and advanced data analytics are essential for real world deployments of such a

system, this study lays the foundation work for implementing such an advanced system on cargo ships.

REFERENCES

- [1] Kamolov, A., & Park, S. (2019). An IoT-based ship berthing method using a set of ultrasonic sensors. *Sensors*, 19(23), 5181. <https://doi.org/10.3390/s19235181>
- [2] Devraj, G., Tharun Kumar, M., Sahith Reddy, C., & Varma, K. S. (2022). IoT enabled shipping containers with location tracking and environment monitoring. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*. Retrieved from <https://www.ijraset.com/research-paper/iot-enabled-shipping-containers-with-location-tracking>
- [3] Chaitra, H. V., Ashika, H. R., Kavitha, K. S., & Keerthika, A. (2021). Smart ship: Using IoT to prevent maritime accidents. *International Journal of Creative Research Thoughts*, 9(1), 1424. <https://www.ijcrt.org/papers/IJCRT2101424.pdf>
- [4] Ahmed, M., Khan, S., & Ahmed, M. (2020). Wireless sensor networks in maritime environments: A survey. *Sensors*, 20(10), 2805.