**Sustainable Waste Management Solution: ESP32-Powered Smart Bin**

**Abstract**

In this project, we present an Internet of Things (IoT) Smart Dust Bin that utilizes the ESP32 and Arduino IoT Cloud to detect and separate dry and wet wastes. The smart dust bin is equipped with different sensors (ultrasonic HC-SR04, Moisture Sensor, SG90 Servo, MG90 Servo) that can detect the type of waste being deposited and classify them depending upon the moisture level of the waste into either dry or wet. The data collected by the sensors is then transmitted to the cloud where it is processed and used to trigger the appropriate waste separation mechanism. This system allows for efficient and automated waste management, reducing the burden on waste collection at processing facilities. Additionally, the smart dust bin can provide real-time data on waste types, enabling more efficient waste management strategies. Overall, this project demonstrates the potential for IOT technology to improve waste management and reduce environmental impact.

**Keywords***—*ESP32, ArduinoIOTcloud, MG90 servo, Arduino IDE, Moisture sensor, ESP drivers.

1. **INTRODUCTION**

In recent years, the rapid growth of urban populations and the corresponding increase in waste generation have posed significant challenges to traditional waste management systems. To address these challenges, there is a growing interest in leveraging the potential of Internet of Things (IoT) technologies to create efficient and intelligent waste management systems. This research paper introduces an innovative approach to waste management through the development and implementation of an IoT-driven system, specifically focusing on an ESP32-based smart dustbin.

The primary objective of this study is to design and deploy a smart dustbin that utilizes the ESP32 microcontroller, which serves as a core component for data sensing, processing, and communication. By integrating IoT capabilities into the waste management infrastructure, this system aims to enhance garbage collection efficiency, optimize resource allocation, and contribute to sustainable urban development.

The smart dustbin using IoT technology uses moisture sensors to detect the presence of liquid or wet waste and separate it from dry waste. By integrating a moisture sensor with an ESP-32 microcontroller, the system can detect the moisture level and make a decision to separate wet from dry waste. When the moisture level crosses a certain threshold, it separates accordingly. Data collected by the moisture sensor is synced with the Arduino IoT Cloud platform, which allows for remote monitoring and management of the system. The cloud platform can store data on the moisture level over time, and provide visualizations of the data to give insight into patterns of usage and waste generation. Additionally, this data can be used to optimize garbage collection schedules and help city officials make better decisions about waste management and disposal. With the rapid increase in the rate of population, there is an increase in garbage which has to be effectively managed to reduce harmful diseases. We present an Internet of Things Smart Dust Bin that utilizes the ESP32 and Arduino IoT Cloud which is used to automatically separate dry and wet wastes and also detect a person and open the lid automatically. The smart dust bin is equipped with sensors (ultrasonic HC-SR04, Moisture.

Sensor, SG90 Servo, MG90 Servo) that can detect the type of waste being disposed of and have equipped a mechanism where we will capture the data through sensors and process it on the cloud to identify the level moisture of the waste, this system allows for efficient and automated waste management, reducing the burden on waste collection at processing facilities. Additionally, the smart dust bin can provide real-time data on waste types, enabling more efficient waste management strategies. Overall, this project demonstrates the potential for IoT technology to improve waste management and reduce environmental pollution

The paper is organized as follows:

* Section 2: Literature Review - Provides a summary of previous research conducted in the area of social media and adolescent mental health.
* Section 3: Methodology - Explains the operation and functioning of our suggested model.
* Section 4: Results - Presents the findings and results obtained from the conducted experiment.
* Section 5: Conclusions - Summarizes the key conclusions drawn from the project and its implications.

1. **LITERATURE REVIEW**

This section provides reviews on various use cases of the Internet of Things which helped in building this project.

In their work, Chokemongkol Nadee et al. [1] introduced a systematic method for mitigating the risk of falls in areas where CCTV surveillance is not available. They utilized ultrasonic sensors placed strategically on the ceiling and walls of the room to detect potential falls. By analysing the threshold signals and comparing the signals from the top and side sensors, the researchers were able to identify specific actions such as standing, sitting, and falling.

This study by HusamKareem et al. [3]. presented a comparison between two widely used development boards for embedded systems design and a newer board, the ESP32, specifically designed for high-performance and cost-effective embedded systems. The comparison includes an evaluation of the hardware specifications and programming tools provided by each board. As a practical demonstration of utilizing the ESP32 module in low-cost and low-energy embedded systems design, a Wi-Fi analyser system is designed and implemented. Additionally, a Graphical LCD is incorporated into the design as an output device.

The study conducted by O. E. Amestica [4] focuses on comparing the execution time of different mathematical operations and a crucial function required for data acquisition or digital control. The comparison specifically evaluates the following aspects: digital port writing, analog signal acquisition, execution of mathematical operations in both integer and floating-point formats and execution of the data processing code. To ensure accurate measurement of execution time, a straightforward methodology is employed, which can be readily implemented for any other digital device.

In their study, Cyprian N. Oton et al. [5] introduced a cost-effective and open-source IoT-based SCADA system for a rural Base Transceiver Station (BTS) site. The system incorporates ESP32 and Arduino IoT Cloud as essential components, offering a solution that is both affordable and capable of facilitating supervisory control and data acquisition in rural BTS environments.

In the study, Madala Kranthi et al. [6] discussed about the ways for designing a smart water flow meter. They gave a brief idea to create water supply system that is more relatable, straightforward , and technologically advanced, reducing manual efforts and ensuring efficient water distribution.

In the study, Dr G Ramesh et al. [7] introduced the importance about the health care system using IoT. They used IoT as a medium to extract the information about the therapy dialysis, conditions, and analyse the neural system responses.

1. **METHODOLOGY**

This section provides the methodology to build the project of creating a Smart Dustbin. Figure 1 shows the conceptual framework that is followed to complete this project.

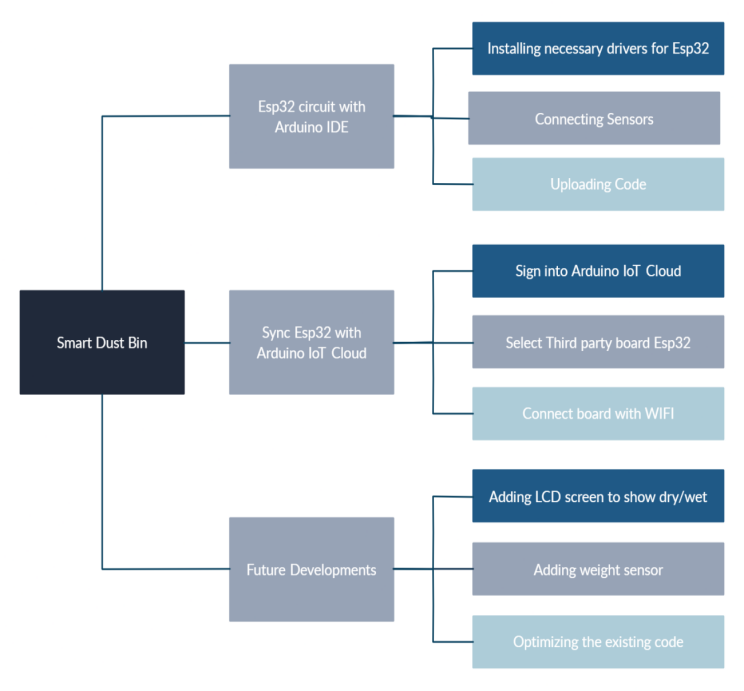


Figure-1: Conceptual framework used for this project

We followed the conceptual framework as shown in Figure-1 to complete this project. Analysis and requirements gathering helped us recognize the needs of the project. The working methodology of the project is discussed further.

The project works on the following algorithm.

**Algorithm:**

1. Begin
2. Checks if the object is closer than 30cm to the ultrasonic sensor
3. Then opens the lid if the object is closer to the bin (30cm).
4. Then it checks the moisture level of the waste.
5. If the moisture level is less than 3000 it segregates into wet waste, else into dry waste
6. After 30 seconds the lid closes.
7. End
   1. **Microcontroller Used:**

This project is based on a microcontroller called ESP-32,

These are the key specifications of that microcontroller,

1. Powerful Microcontroller: The ESP32 is a high-performance microcontroller, enabling simultaneous execution of multiple tasks, making it suitable for real-time processing applications such as IoT devices and robotics.
2. Wireless Connectivity: With built-in modules, the ESP32 can communicate wirelessly with other devices, facilitating remote monitoring and control for applications like smart homes and industrial automation.
3. Versatile Set of Peripherals: The ESP32 has a rich set of peripherals, including UART, SPI, I2C, PWM, and ADC interfaces. This makes it easy to connect to other devices and sensors and enables it to perform a wide range of tasks.
4. Meagrely in Consumption: The ESP32 is developed for low power consumption, making it perfect for battery-powered devices. It also includes a power management unit (PMU) that allows it to enter different power modes, depending on the application requirements
5. Programmable with Arduino IDE: This board can be programmed using the popular Arduino IDE, which makes it easy to get started with and to develop applications quickly.
   1. **Connecting Esp-32 To Components**

The ESP32 microcontroller is utilized in conjunction with an ultrasonic sensor [1] and a servo motor [2] to create a smart dustbin system. The ESP32, programmed using the Arduino language, reads data from the ultrasonic sensor to determine the distance of an object. When an object approaches within 30cm of the dustbin, the ESP32 sends a signal to the servo motor, opening the bin's mouth for waste disposal. After the waste is dumped, the ESP32 measures moisture using a moisture sensor. Based on the moisture data, the ESP32 directs the waste to either the wet waste section or the dry waste section by sending appropriate signals to the servo motor. This process is facilitated by the program which continuously monitors the distance using the ultrasonic sensor, triggers the servo motor for mouth opening, and determines waste segregation based on moisture sensor readings.

* 1. **Connecting Esp32-with Arduino Iot-Cloud**

[4] To connect the smart dustbin to the Arduino IoT Cloud platform and check the real-time value of moisture sensors, Define the moisture sensor pin and set the threshold value for dry and wet waste, in the program. Read the moisture sensor value in the loop function and store it in a variable.

[5] In the loop function, use the variable to send the moisture sensor value to the cloud platform. You can also use it to send the state of the waste, whether it is wet or dry, based on the threshold values that we set.

On the Arduino IoT Cloud platform, the following steps should be taken to create a new widget that displays the real-time value of the moisture sensor and the waste state:

1. Access the Arduino IoT Cloud platform and log in to your account.
2. Navigate to the project or device where the moisture sensor and waste state data will be monitored.
3. Look for the option to add a new widget or component to the project's dashboard.
4. Select the appropriate widget type for displaying real-time data, such as a gauge, line chart, or value display.
5. Configure the widget properties, including the data source and visualization settings.
6. Choose the moisture sensor as the data source for one part of the widget, ensuring that it is set to display the real-time moisture sensor readings.
7. Select the waste state as the data source for the other part of the widget, ensuring that it reflects the real-time state of the waste (e.g., empty, partially filled, full).
8. Save the widget configuration and add it to the project's dashboard.
9. Once added, the widget will display the real-time value of the moisture sensor and the state of the waste, providing you with remote monitoring capabilities.
10. By analysing the collected data, you can gain insights for better waste management, including waste segregation, scheduling of collection, and understanding the usage patterns of the bin.

With this setup, you can remotely monitor and analyse the moisture sensor data and waste state in real time, enabling more efficient waste management practices and informed decision-making.

* 1. **Arduino Program for Esp-32 Without Connecting to Cloud**

Code\_start

Visit Reference - Arduino program for Esp-32 without connecting to the cloud [6]

Code\_end

* 1. **Arduino program for Esp-32 with Cloud connection**

Code\_start

Visit Reference - Arduino program for Esp-32 with Cloud connection [7]

Code\_end

* 1. **Connections and Working Model:**



Figure-2: Dustbin front view



Figure-3: Top view

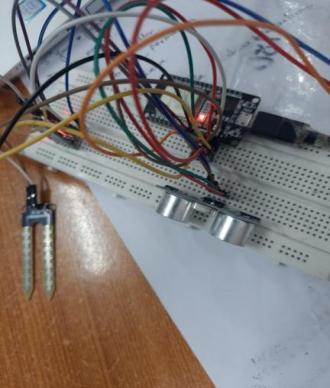


Figure-4: Connection

* 1. **Arduino IOT Cloud:**

It is an intuitive platform created by Arduino to facilitate the connection of IoT devices to the internet and enable remote management. It simplifies the development of IoT applications by handling complex tasks such as network protocols, security, and data management, freeing developers to focus on building powerful and robust applications.

Here are some key features of the Arduino IoT Cloud:

1. Device Management: The Arduino IoT Cloud provides a central dashboard for managing connected devices. Users can monitor the status of their devices, update firmware, and configure settings remotely.
2. Data Management: The platform provides a simple way to collect, store, and analyze data from IoT devices. Users can create custom dashboards, charts, and graphs to visualize data and gain insights into device performance.
3. Security: The Arduino IoT Cloud is designed with security in mind. It uses industry-standard encryption and authentication protocols to ensure that data is transmitted and stored securely.
4. Integration with Other Platforms: The platform can be integrated with other cloud services, such as AWS, Microsoft Azure, and Google Cloud, to provide additional functionality and scalability.
5. Easy to Use: The Arduino IoT Cloud is easy to use and requires no programming skills or knowledge of networking protocols. It provides a simple drag-and-drop interface for configuring devices and setting up dashboards.
6. **RESULTS**

Increased efficiency in waste management: The smart dustbin would allow for real-time monitoring of the segregation of waste, which would enable sanitation workers to optimize their routes and schedules and collect waste more efficiently. By separating dry and wet waste, the smart dustbin would promote proper waste segregation and help to reduce the amount of waste that ends up in landfills. Increased transparency and, by syncing the data from the moisture sensor with the Arduino IoT Cloud platform, officials would be able to monitor the usage and waste generation patterns over time, and make better decisions about waste management and disposal.

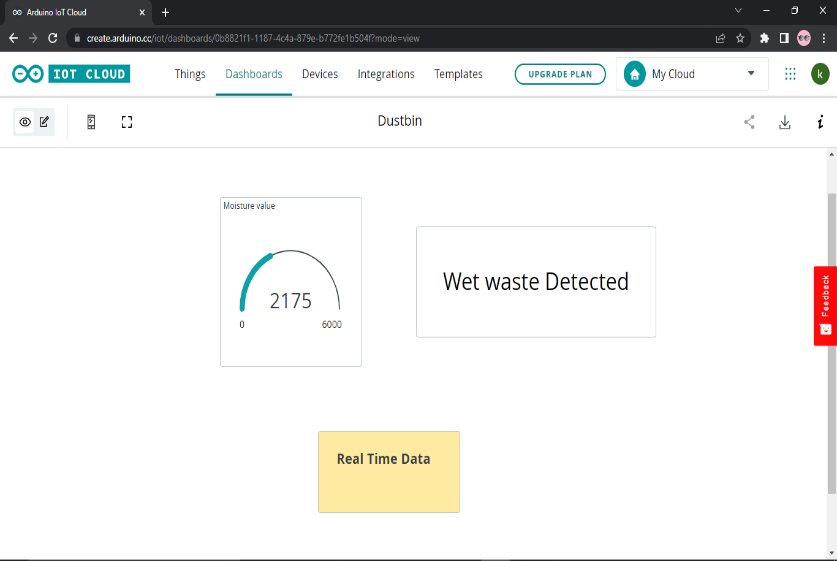


Figure-5: Arduino IoT Cloud Website Design

****

Figure-6: Arduino IoT Cloud Mobile Application Design

1. **CONCLUSIONS AND FUTURE SCOPE**

With the help of the proposed system, one can easily segregate the waste into wet and dry. This proposed model is cost-effective and reliable. Implementation of this requires a microcontroller and some sensors. The project can be further improved and many more features can be added such as reorganizing how much the bin has been filled. However, with the help of the current project, one can segregate the waste easily within his house. This project has a lot of potentials to be improvised further and add many more improvements. This system can also find large market potential by which this idea can be commercialized.

One potential future improvement for the above smart dustbin project would be to add an LCD display to show the type of waste being deposited, whether it is dry or wet, and to include a weight/pressure sensor to calculate the weight of the waste deposited, this will give more insights into the pattern of waste generated and also on how much capacity is remaining in the bin before it gets full. This will also help to increase the transparency and data-driven decision-making, with the added information on the weight of waste. It can also aid in sending the signal to the collection trucks at the right time to collect the waste before it reaches its full capacity.

1. **REFERENCES**

# [Chokemongkol Nadee](https://ieeexplore.ieee.org/author/37085531043), [Kosin Chamnongthai](https://ieeexplore.ieee.org/author/37269457800) Ultrasonic array sensors for monitoring of human fall detection(2015) **DOI:**[10.1109/ECTICon.2015.7207097](https://doi.org/10.1109/ECTICon.2015.7207097)

# [Fudao Sun](https://ieeexplore.ieee.org/author/37088957303), [Jian Luo](https://ieeexplore.ieee.org/author/37085802788) Design and Research on Miniaturization and Lightness Servo Motor Controller(2021) DOI: [10.1109/AEECA52519.2021.9574233](https://doi.org/10.1109/AEECA52519.2021.9574233)

# [Husam Kareem](https://ieeexplore.ieee.org/author/37086528377); [Dmitriy Dunaev](https://ieeexplore.ieee.org/author/38244798900)The Working Principles of ESP32 and Analytical Comparison of using Low-Cost Microcontroller Modules in Embedded Systems Design(2021) DOI: [10.1109/ICCSS51193.2021.9464217](https://doi.org/10.1109/ICCSS51193.2021.9464217)

# [O. E. Amestica](https://ieeexplore.ieee.org/author/37087500609); [P.E. Melin](https://ieeexplore.ieee.org/author/37396960200); [C.R. Duran-Faundez](https://ieeexplore.ieee.org/author/38277448300); [G.R. Lagos](https://ieeexplore.ieee.org/author/37087502356) An Experimental Comparison of Arduino IDE Compatible Platforms for Digital Control and Data Acquisition Applications(2019) DOI: [10.1109/CHILECON47746.2019.8986865](https://doi.org/10.1109/CHILECON47746.2019.8986865)

# [Cyprian N. Oton](https://ieeexplore.ieee.org/author/37088749353); [M. Tariq Iqbal](https://ieeexplore.ieee.org/author/37286201800) Low-Cost Open Source IoT-Based SCADA System for a BTS Site Using ESP32 and Arduino IoT Cloud(2021) DOI: [10.1109/UEMCON53757.2021.9666691](https://doi.org/10.1109/UEMCON53757.2021.9666691)

* 1. Madala Kranthi, Velagapudi Sreenivas, K. Prabhakar, G. Ramesh “A Novel Approach of Smart Water Flow Meter using IOT”, 3rd International Conference on Design and Manufacturing Aspects for Sustainable Energy (ICMED 2020), Volume 309, September, 2021
  2. G. Ramesh, et al., Smart Medical Assistance using Soft Computing, E3S Web Conf., 430 (2023) 01189, DOI: <https://doi.org/10.1051/e3sconf/202343001189>.
  3. [Arduino program for Esp-32 without connecting to cloud](https://drive.google.com/file/d/1JXWaASnxpZTu2CmutpRf5GQjhboZR6Wz/view)
  4. [Arduino program for Esp-32 with Cloud connection](https://drive.google.com/file/d/1sRePMbFsK-ArRcwUYpv6_0e0RNldb2A9/view)