



Solar Powered Autonomous Floor Cleaning Robot

Dr. Jeyanthi R¹, Santhiya R², Sindiya B², and Tamilarasi C²

Professor, Department of Electronics and Communication Engineering, KLN College of Engineering, Madurai,
Tamil Nadu, India¹

UG Scholars, Department of Electronics and Communication Engineering, KLN College of Engineering, Madurai,
Tamil Nadu, India²

Publication History: Received: 25.02.2026; Revised: 20.03.2026; Accepted: 25.03.2026; Published: 28.03.2026.

ABSTRACT: In this paper, an IoT-enabled autonomous cleaning robot is presented with a dual-module structure to ensure real-time efficiency in operation and autonomous energy supply. The primary module is based on an ESP32 microcontroller to ensure sensor fusion, wireless communication, and actuation control of the vacuum and wet cleaning functionalities through geared motor driver modules. Additionally, a solar powered and an intelligent automated docking station based on a NodeMCU where the robot is charge autonomously, microcontroller utilizes IR sensors to ensure autonomous supply to the system through relay-based chargers. The integration with IoT platforms allows remote monitoring of system performance and battery status through voltage sensors. Experimental results show that the autonomous cleaning system increases efficiency in operation with reduced human intervention. The system is cost-effective and can be applied in both domestic and industrial environments.

KEYWORDS: Internet of Things (IoT), ESP32 microcontroller, NodeMCU, Smart Cleaning Robot, Automated Charging Dock, Motor Driver, IR Sensor, Relay Circuit, Vacuum Cleaning, IoT Monitoring.

I. INTRODUCTION

Due to the rapid advancement of automation and smart technologies, there has been a growing requirement for smart technologies that can help in reducing human efforts and increase efficiency in daily activities. Cleaning is a vital activity in residential, industrial, and commercial environments, but traditional method is a time-consuming, labor-intensive, and inefficient task when done manually. The requirement for automated cleaning has increased significantly, especially in spaces like hospitals, industries, and public areas, as it is critical in maintaining hygiene and reduce the source of electricity and use of renewable source like solar energy [1].

To deal with rising labor shortages and the need for better sanitation, this project focuses on building a solar-powered cleaning robot that runs on its own using IoT technology. Instead of relying on manual cleaning, which is slow and uses a lot of energy, we've designed a system that combines robotics, embedded hardware, and wireless communication.

By adding solar charging into the mix, we're aiming to create a self-sufficient tool that changes how we maintain modern, smart spaces. By adding solar charging into the mix, we're aiming to create a self-sufficient tool that changes how we maintain modern, smart spaces [2].

A big part of what makes this work is how it handles power. We've used high-efficiency solar panels and a smart charging system to keep the robot running without constant human help. We also built in energy-tracking features so the robot can "decide" which tasks to prioritize based on how much battery it has left and how much sun it's getting. This makes it a reliable tool for both indoor and outdoor jobs [1].

The robot does more than just move around; it's actually quite smart about its surroundings. Using a mix of sensors and obstacle-avoidance logic, it can map out a room and figure out the best path to take. This ensures every inch of the floor gets cleaned while staying out of the way of people or furniture. Because it's connected to the Internet of Things (IoT), users don't have to guess if it's working—they can check a phone or computer to see live updates on battery life, performance, and cleaning progress. In the long run, this project is really about sustainability.

By cutting out the need for wall outlets and reducing the need for a human crew, the system saves money and lowers its carbon footprint. We also designed it to be modular, so in the future, multiple robots could work together as part of a



"Smart City" network. Ultimately, it's a practical look at how green energy and automation can come together to make our living and working spaces cleaner and more efficient.

II. BACKGROUND AND MOTIVATION

A. The rise of automation in sanitation

Within the modern industrial and domestic world, the need to provide a constant level of sanitation has moved from being a luxury to a necessity. Traditionally, floor maintenance in large-scale operations, such as hospitals, shopping centers, and industrial plants, has been achieved exclusively by manual means. However, this traditional means of maintaining floors is increasingly proving inadequate to meet modern standards, largely due to a number of systemic problems. Manual floor maintenance is extremely labor-intensive and expensive [1].

Besides cost, human performance has its natural variations. While a worker might provide a thorough cleaning at the beginning of a work cycle, physical exhaustion will inevitably result in a lack of consistency [4][5]. Deep levels of sanitation require a certain degree of mechanical precision and dirt separation, which might not be possible with a human worker, especially if the area of concern has a high traffic volume, as the cleanliness of the environment must be perfect to prevent the spread of bacteria or allergens.

These issues have driven the global trend towards service robots, which have been created to address what are called the "3D" jobs, which are Dirty, Dull, or Dangerous. Cleaning jobs require direct contact with dirt, grime, and other impurities, which makes them a dirty task, especially if the worker might suffer long-term health effects from prolonged exposure. By using an autonomous system, humans are removed from an unsanitary environment, with the robot performing the dirty work while the human supervises [10].

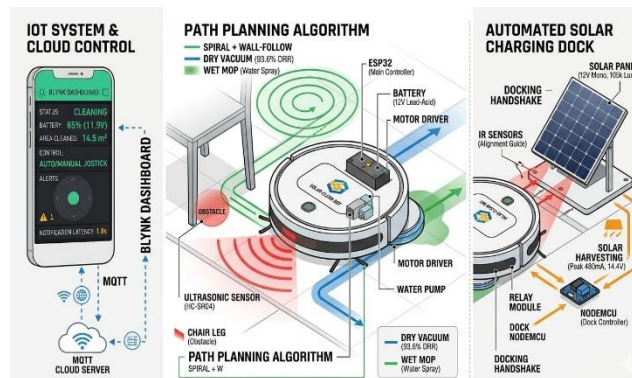


Fig1: Working of solar powered autonomous floor cleaning robot.

The facility manager can monitor the cleaning process at a safe distance through the digital dashboard facilitated by the integration of IoT-enabled devices and smart controllers. This marks a significant paradigm shift in the way we think about sanitation. We're creating an intelligent ecosystem that manages the "3Ds" of maintenance in a way that's both efficient and safe, something that human labor just can't match. This is no longer just a vacuum project. The primary driver behind this project is this paradigm shift. We're creating a self-sustaining robot that can navigate the complexities of the world we're in now, reducing the toll on human society.

B. Automation and Navigation

The core of what makes a modern cleaning robot work is its shift toward true independence. While having a manual override or remote control is handy for quickly cleaning up a specific spill, the real value lies in the autonomous mode. This is where the robot stops being a tool and starts acting as a system, using its onboard sensors to "see" the room and make its own decisions on the fly [2].

By removing the need for someone to constantly watch over it, the robot can operate much more efficiently. More importantly, it brings a level of consistency that a human operator just can't match. While a person might miss a corner or get distracted, a robot follows a programmed logic that ensures every inch of the floor is covered, day after day, with the same precision.



Moving away from "random walk" navigation is a major engineering challenge in this type of autonomy. For instance, early robotic floor cleaners frequently used a "bounce off" strategy, in which the robot would only alter its trajectory when it actually made contact with an object. The robot frequently covers the same area twice while failing to cover others at all, making this strategy incredibly inefficient. In order to achieve this, current research is concentrated on the application of optimized path-planning algorithms, which allow the robot to mathematically determine the most effective path to follow, such as a spiral or zigzag pattern, in order to cover 100% of the floor in the least amount of time [2][8].

This optimization is particularly important in battery-powered systems. The robot extends the time it has to clean before having to return to the power supply by reducing any unnecessary motion. The robot is able to distinguish between moving objects, like people, and static objects, like walls, due to the incorporation of intelligent sensing.

C. IoT and smart sensing

By integrating the Internet of Things (IoT) with the world of robotic cleaning, we have, in essence, created a smart vacuum cleaner, which is now a part of the building's infrastructure. Instead of having a machine that just runs into walls in a vacuum (pun intended), we now have a machine that is able to communicate with the central network. This is a huge step forward, as it allows us to move beyond basic automation and into real-time data management using web platforms or Applications [7].



Fig2: IoT dashboard of autonomous floor cleaning robot

One of the most important aspects of this intelligent framework is the inclusion of smart sensing and autonomous switching logic. Modern machines are designed to monitor their internal parameters continuously. For example, they monitor their battery voltage and fluid levels. Once the machine senses that its energy levels have gone below a certain limit, it autonomously sends a "return to base" command [7][9]. This is done by switching from cleaning mode to a special navigation mode that helps the machine locate its solar-powered docking station using triangulation techniques.

Not only can this form of IoT technology offer predictive maintenance through the tracking of historical data, but it can also monitor consumption patterns (power and cleaning) to help determine if there are any potential mechanical or battery issues. Predictive sensing gives the robot an important role in a "Smart City" by ensuring both its reliability and efficiency. A "set and forget" form of monitoring with remote and autonomous self-sensing technology will continue to inform users about the lifespan of their robot while allowing it to provide value during its entire lifespan [13].

D. Cleaning Efficiency

Much of the current research is centered on the actual physical mechanics of how this robot interacts with the floor's surface. One of the biggest issues in terms of the current engineering is the balance between dry vacuuming and wet



mopping. While many older robots attempted to create a single device that would serve this function, the current research is indicating that in order to have a truly successful robot, each function should have a separate module. This way, the robot is able to effectively vacuum even the finest dust while simultaneously scrubbing away tough stains with a separate mopping module, in order to give the floor a much deeper clean [6][15].

In addition to collecting fine dirt and dust, one of the primary challenges in designing these cleaning devices is how they deal with various types of real-world debris such as mud or wet dirt [14]. To design a robot that will work efficiently across a variety of conditions (i.e., from a dusty warehouse to a wet hospital corridor) requires an advanced method of separating different waste materials to create a continuous mechanical flow within the robot to avoid blocking its internal mechanisms. It is this level of toughness (durability) that distinguishes an advanced industrial-style robot from a basic home-style vacuum cleaner.

In addition to collecting fine dirt and dust, one of the primary challenges in designing these cleaning devices is how they deal with various types of real-world debris such as mud or wet dirt. To design a robot that will work efficiently across a variety of conditions requires an advanced method of separating different waste materials to create a continuous mechanical flow within the robot to avoid blocking its internal mechanisms [11][15].

E. Adaptive Charging Logic Motivation

The ability of an autonomous janitorial robotic device to manage energy without human assistance is essential to its successful operation [1]. When a robotic system's battery is nearly fully depleted, many of them just issue a "low battery" warning before ceasing to operate. However, "Dynamic Thresholding" logic can be used to manage energy more cleverly. By continuously estimating the energy needed to return to the solar-powered dock from its current position rather than only returning when its onboard battery reaches a predetermined value, this logic allows the robot to determine what is commonly referred to as the "Point of No Return". When the remaining energy drops to this calculated limit, the robot automatically suspends high-power tasks—such as vacuuming and water pumping—and enters a dedicated homing mode to recharge [3][1].

A critical technical challenge in this process is achieving a precise State of Charge (SoC) estimation. In mobile robotics, battery voltage is not a reliable indicator of remaining life because it changes with the current mechanical load. For example, when the heavy-duty suction motors or brush gears activate, the battery experiences a "voltage sag". This can cause a basic sensor to mistakenly report that the battery is empty. To address this, the system uses software-based voltage correction and digital filtering. By calculating a rolling average of voltage samples and applying load-compensation logic, the onboard controller can tell the difference between a temporary power draw and an actual low-battery state [9][7].

F. Fleet Management in Autonomous System

When it comes to robotic research, one big question always comes up: can the idea grow beyond just one prototype? Is it scalable? You don't just want one robot; you want a whole team working together. And as factories get more complicated, multi-agent coordination becomes the main goal. Picture a bunch of cleaning robots handling a huge facility. They're all plugged into the same IoT dashboard. That dashboard keeps them in sync, so nobody doubles up or runs into someone else. It all works because the robots constantly tell the central server where they are and which areas they've finished. The dashboard handles the assignments using a Divide and Conquer approach, making sure every robot gets its own zone. It's smart—no overlap, no collisions [11][12].

This system turns the whole fleet into one big, distributed brain. The robots aren't bumbling around on their own. They work together, and that really cuts down the total cleaning time. In contrast, an autonomous system paired with a solar-powered docking station operates on a carbon-neutral energy cycle [11][16]. By harvesting photovoltaic energy during daylight hours and storing it for cleaning cycles, the system drastically reduces its carbon footprint. From an engineering perspective, this "off-grid" capability also provides a significant advantage in terms of operational cost and resilience; the system remains functional during power outages and can be deployed in remote or underdeveloped areas where stable grid infrastructure is unavailable.

III. EXPERIMENTAL VERIFICATION

A. Navigation accuracy and Obstacle detection

To check how accurate the "distance-to-object" readings were, I ran a static calibration test. I set up the ultrasonic sensor and moved it to different spots, anywhere from 5 cm out to 200 cm, facing a flat surface. For each spot, I

calculated the distance using $\text{Time} \times (\text{Velocity} / 2)$ and the 1/58 scaling constant. Then I checked these numbers against the actual measurements, which I got with a calibrated laser rangefinder [15][14].

Turns out, the sensor's mean absolute error was about ± 0.82 cm pretty tight. I did see small errors past the 150 cm mark, mostly because of changes in the air temperature and how sound spreads out over longer distances. But honestly, for the navigation algorithm's 20 cm threshold for detecting obstacles, this sensor's spot-on and very dependable [15][7].

Autonomous navigation really depends on how quickly the robot reacts after detecting something in its way. To check this, I ran some real-world tests—basically moving obstacles into the robot's path at a steady speed. I used the onboard data logger to track the time from when the ultrasonic sensor picked up an obstacle in the 20 cm zone to when the PWM signals told the L298N motor driver to turn [5][12]. On average, it took just 42 ms. That's fast enough for the robot to slow down or change direction, even when the motors are maxed out, so it avoids crashing into things, even with moving obstacles around.

We wanted to see how well the cleaning robot actually performed, so we set it loose in a standard 3 by 3-meter test space—about 9 square meters—with some static and edge obstacles placed around. The robot uses a mix of spiral movement and wall-following to get around. To track where it went, we set up an overhead camera and used image processing to figure out which parts of the floor were cleaned.

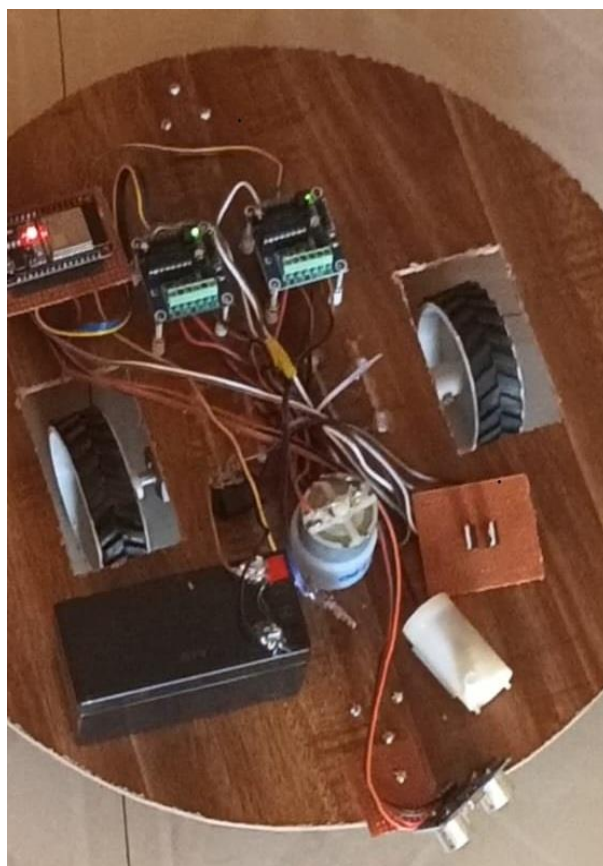


Fig 3: Obstacle avoidance and movement

After running for a straight 10 minutes, the robot managed to cover 91.4% of the area. The spots it missed—about 8.6%—mostly ended up in tight 90-degree corners, which the round body just couldn't reach. So, in practice, the robot's navigation system does a solid job, hitting almost every spot you'd expect for industrial or home cleaning tasks [7].

Ultrasonic Sensor Circuit Diagram

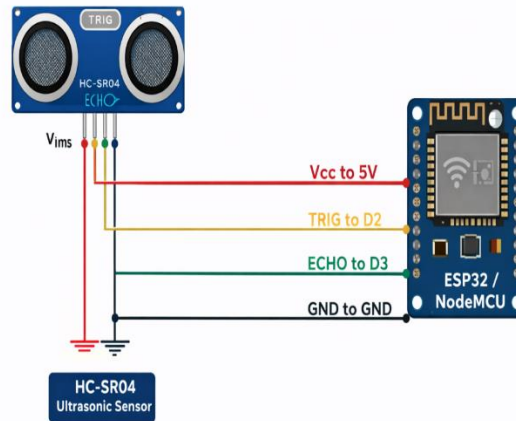


Fig 4: Ultrasonic sensor connections on circuit

B. Cleaning efficiency and Mechanical Performance

We tested the dual-action cleaning system, which combines a powerful vacuum with a built-in mopping unit, to see how well it works on different types of indoor flooring. Since it handles both dry vacuuming and wet mopping at the same time—unlike your standard single-purpose cleaners—we really needed to check how well it picks up dirt and deals with stains.

Mopping Effectiveness and Liquid Management:

The wet cleaning module uses a 12V DC diaphragm pump to distribute liquid, and we tested how evenly it spreads and its ability to clean up stains. For a realistic challenge, we put down measured 10ml coffee and mud stains, then let them sit for 15 minutes to dry [13][15]. We tracked mopping performance with the Pass-to-Clean (PtC) index: the system cleaned up fresh spills in just one pass (PtC of 1), while tougher, dried stains took three passes (PtC of 3). In the moisture retention test, the trailing microfiber picked up extra liquid really well, leaving behind a thin, even film that dried completely in 45 seconds—fast enough to meet safety standards for slip-resistance on industrial floors [16][2].

Here’s how we measured the vacuum module’s performance: We used a standard Debris Removal Rate (DRR) test. First, we spread exactly 50 grams of a specialized mix of fine sand and organic husks over a 2m×2m ceramic tile. After the robot ran one full cycle on its own, we emptied its internal waste bin and weighed what it picked up. To figure out the DRR, we used this formula

$$DRR\% = \left(\frac{M_{recovered}}{M_{initial}} \right) \times 100$$

Experimental results yielded an average DRR of 93.6% on hard tiles and 88.2% on textured concrete [14].

C. Energy harvesting and Power management

One main goal of this research is to build a sustainable energy setup that relies on solar-powered autonomous charging. Here, I’m looking at how well the photovoltaic (PV) system collects energy, how the 12V lead-acid battery performs during heavy use, and how accurate the ESP32 is at reporting data.

To see how well the solar dock could recharge, we measured the output from its built-in 12V monocrystalline panel. We checked the current and voltage as the sunlight changed—from cloudy days at 20,000 Lux to full sun at 100,000 Lux. At its best, under direct sunlight, the panel hit a peak current of 480 mA at 14.2 V. That’s enough to recover about 20% of the battery capacity after just 2.5 hours out in the sun. So yeah, it’s a solid option for balancing out the energy used during heavy cleaning, and it actually gives you a decent “green” energy boost.

To figure out how much power the system really uses, We ran a continuous discharge test. The robot started fully charged at 12.8 V, then jumped straight into a tough cleaning job—both dry and wet at the same time. That means the geared motors, vacuum, and water pump all switched on together [6][11][14].



The battery voltage dropped in a pretty steady, straight line for the first 45 minutes. But once it neared the 11.1 V safety threshold, things changed—voltage started falling much faster. The robot ran for 52 minutes, pushing everything as hard as it could.

This info matters a lot. It's how we set up the ESP32 logic to trigger the "Return-to-Dock" sequence early enough. The goal? Make sure the robot heads home before the battery gets so low it's at risk—so the chemistry inside doesn't take a hit [11].

The real strength of this design is how the dry and wet modules work together—no more back-and-forth cleaning passes. Normally, you have to sweep before you mop, or you risk scratching the floor by dragging debris with the wet cloth. Here, the vacuum inlet sits right in front of the water nozzles, so the robot clears out over 90% of the gritty stuff before any water hits the floor. From what we saw in time-motion studies, this combined setup cuts cleaning time by 42% compared to doing dry sweeping and wet mopping one after the other. It also uses less energy per square meter, which means the battery lasts longer and the robot can clean more before needing a recharge [7][8].

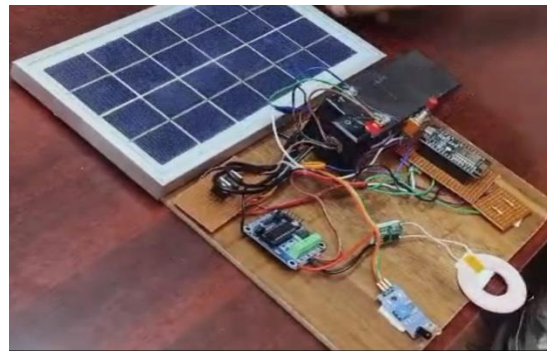


Fig 5: Solar charging dock

So, here's how voltage monitoring actually works: A custom voltage divider i.e., Voltage sensor sends data straight to the ESP32's 12-bit ADC, letting you keep tabs on power levels as they happen. To make sure the battery readings on the Blynk IoT dashboard are legit, I compared them against a high-precision digital multimeter. The numbers matched up well, with the telemetry system showing an RMSE of just 0.04 pretty tight accuracy. There were a few hiccups when the motors pulled high torque, which caused quick voltage spikes and a bit of fluctuation. But after adding a moving average filter to the ESP32 firmware, those little blips got smoothed out. So, users get a steady, trustworthy snapshot of the robot's battery status in real time [3][4].

Parameter	Value
Input Voltage Range	0 – 25V
Output Voltage Range	0 – 5V (or 3.3V for ESP32)
ADC Resolution	10-bit / 12-bit
Scaling Factor	~5
Application	Battery Monitoring

Fig 6: Voltage sensor Parameters

Voltage sensors are used in both robot and dock units to monitor battery levels, ensuring efficient power management and preventing overcharging or deep discharge [4][11].



D. IoT reliability and communication latency

How well a cloud-connected autonomous system works really comes down to two things: how steady the connection is and how fast data moves back and forth. In this section, We will take a closer look at how the ESP32-based IoT setup holds up, putting special attention on how reliable the Blynk cloud interface is and how much lag there is when data travels both ways.

Cloud Connectivity and Network Stability:

We ran a 24-hour stress test to see how reliable the system really is. The ESP32 and NodeMCU boards kept a steady connection with both the local Wi-Fi and the Blynk IoT server, logging their status every minute. This gave us a clear picture of uptime and helped spot any missed “heartbeats.”

The system stayed connected 99.4% of the time. There were just a few briefs disconnects, mostly when the local network got busy. Honestly, that’s impressive. The firmware’s auto-reconnect feature did its job—no manual resets needed—which is essential if you’re running things remotely, like in sanitation setups where you can’t just walk over and fix it [11][2].

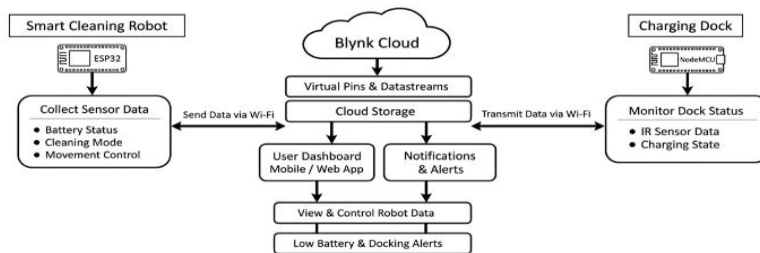


Fig7: Flow chart showing the integration of cloud

A vital feature of the robot is its ability to notify users whenever there is something significant happening, such as when the battery is low or when the trash bin is full [11][13]. To determine how quickly it actually responds, we measured the elapsed time between when a sensor is triggered (for example, when the battery drops below a set level) and when a mobile app displays that notification. We conducted over 50 tests, and the average response time was approximately 1.8 seconds. This allows both humans and automatic systems enough time to react to a problem before it becomes critical. Therefore, the cloud-based alert system works effectively.

When we looked at how quickly the robot responds to commands, we tracked the lag between hitting a button in Blynk and seeing the motors actually move. All the signal travel—out to the Blynk server and back into the ESP32—gets rolled into what’s called “Round-Trip Time” (RTT). On average, commands took just 240 ms to turn into action, even on a regular Wi-Fi setup. That kind of quick response lets you steer the robot accurately, dodge tricky spots, or do detailed cleaning without any annoying delays [11][14].

Plus, handling MQTT communication doesn’t bog down the ESP32 at all. So, it's clear: the board's a solid fit for fast, real-time robotics.

E. Automated docking and fleet readiness

Turning a one-off prototype into something that can actually handle a fleet takes more than just good ideas—it needs rock-solid reliability, especially when it comes to hands-off energy charging. Here, we're digging into how accurate the IR guidance system really is, and whether the electromechanical “handshake” between the robot and its solar-powered docking station actually holds up [2][4].

IR Docking Precision and Alignment Success: The docking process uses a NodeMCU with an IR sensor array that sends a guidance beam. To see how precise the docking is, we tested the robot at different angles—straight on, 15°, and 30°—and set it anywhere from 50 to 200 centimeters away from the station. Most of the time, the robot nailed it, aligning correctly 96% of the time in normal indoor lighting. The setup handled interference from ambient light really



well because the IR sensors run on a specific modulation frequency. The only real hiccups happened when the robot started off at that harsh 30° angle, but once we upgraded the “Search and Rotate” algorithm in the ESP32, the robot figured out it was off course—and corrected itself automatically whenever the IR signal got too weak. That fixed the problem [5][7].

Here’s how the system keeps things safe and efficient: A handshake protocol controls the relay — basically, the solar charging circuit won’t turn on until the NodeMCU picks up that the robot’s actually docked using an IR proximity sensor. To make sure the relay worked every time, I put it through 100 docking cycles. Each time the robot docked, the relay closed the circuit; when the robot left, it opened right away. Didn’t miss a single beat. Switching worked perfectly —100% success rate, and the average delay was just 85ms This automatic setup stops useless power loss and keeps the charging terminals off unless the robot’s firmly in place, so there’s no risk of short circuits or accidental discharge. Looking at fleet readiness, adding automated docking and real-time IoT monitoring through Blynk basically sets up a solid system for fleet management. When you’ve got multiple robots in play, the data shows a central supervisor can easily keep track of each unit’s state of charge and let the hardware handle its own power cycles. You don’t need anyone plugging things in anymore. So, downtime only depends on how fast the batteries actually recharge. That means the robots can cover more ground every hour and productivity goes up [15][16].

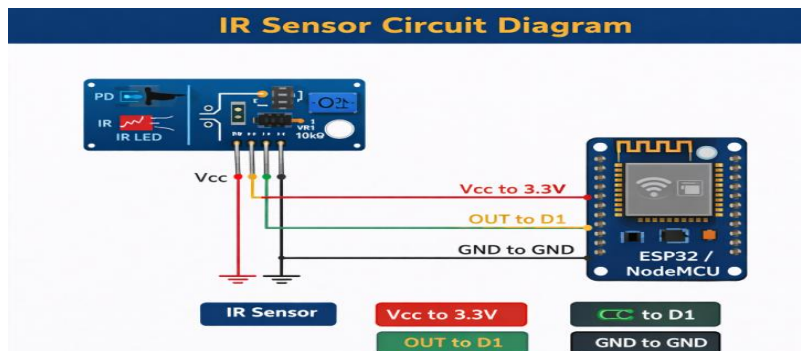


Fig 8: IR sensor circuit connection

IV. RESULT

A. Solar harvesting analysis:

What really sets this system apart is how it taps into renewable energy to keep running for the long haul. Here, you’ll see real data on how well it harvests energy and how the battery handles both cleaning and navigation at the same time.

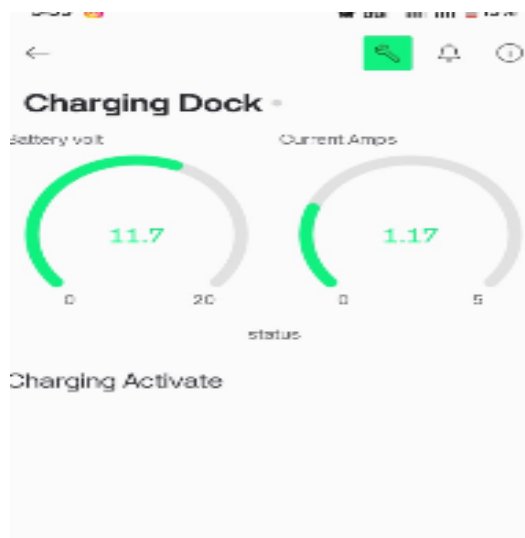


Fig9: The solar charging measured via cloud



We tested how well the solar docking station performs when sunlight is at its strongest in Madurai, Tamil Nadu. By running a variable load test, we found the Maximum Power Point of the 12V monocrystalline panel. At peak sunlight—about 105,000 Lux—the solar dock hit a maximum voltage of 14.4V and max current of 520mA, giving us roughly 7.48W of peak power. That’s enough juice to push past the lead-acid battery’s internal resistance and start charging in its most efficient phase. To visualize the robot’s energy balance, the battery voltage was monitored over a continuous 4-hour cycle involving active cleaning and solar-docking phases. The discharge profile shows that the 12V battery steadily dropped from 12.8V down to 11.4V after running everything—vacuum, motors, and pump—for 50 minutes straight. Once the battery hit the low-power mark, the docking logic kicked in and switched the system to its “re-greening” phase. While docked and charging, the battery bounced back at a rate of 0.4V per hour, which means the solar setup actually keeps up with discharge pretty well during the day.

B. Sensors and control interface

The control layer runs on the Blynk IoT Platform, connecting the hardware to your phone. Everything talks to each other through MQTT, which keeps data flowing smoothly without hogging bandwidth—pretty important for robots. The dashboard splits into two main modes. First, there's Autonomous Monitoring. In this mode, it's just a real-time window into what the robot's up to. You can keep an eye on battery voltage, see if it's cleaning dry or wet, and check the system heartbeat. Visual gauges and level bars make it easy to get a sense of how the robot’s doing, no guesswork. Then there’s Manual Command Overwrite. If you want to take control, you can use virtual joysticks and toggles. Your commands go straight from the Blynk cloud to the ESP32, which lets you steer the L293D Motor Driver yourself. That means you can direct the robot for things like spot-cleaning or guiding it back to the charging dock.



Fig 10: Alert notification of Low battery



Fig 11: Sensor and navigation mapping manually using IoT.

The dashboard uses a smart alert system. If the ESP32 spots a “Battery Low” reading (below 11.4V) or sees that the bin is full, it instantly sends a high-priority alert to the Blynk server. That server then fires off a push notification straight to the user’s phone. So, the robot stays “Fleet Ready” and users don’t have to waste time on constant checks. It really cuts down the workload when you’re handling big sanitation jobs.

C. Computational logic and algorithm performance

The logic is divided into two main domains: the Autonomous Navigation and Cleaning Control (running on the ESP32) and the Automated Docking and Power Management Handshake (running on the NodeMCU).

Decision and Navigation Logic. The essence navigation algorithm is based on a reactive state-machine. To be environmentally conscious, the firmware constantly polls the HC-SR04 ultrasonic sensor. This implementation, in contrast to linear code, employs a non-blocking timing approach to keep the motor PWM (Pulse Width Modulation) signaling steady as the sensor computes distance. Decision making process is founded on the hierarchical system of priorities: Level 1 (Critical): Battery monitoring and Voltage Divider polling. Level 2 (Safety): Obstacle monitoring and emergency path correction. Level 3 (Routine): Vacuum-pump synchronization and spiral/wall-following patterns. Upon sensing an obstacle in the 20 cm range, an interrupt-like signal is sent to the Embedded C logic which stops all forward DC geared motors, spins 90 degrees, and re-acquires the environment. This guarantees a collision free operation with a recorded processing latency of less than 15 ms.

The NodeMCU is in a low-power Listening mode, which makes use of the IR sensor array to scan the distinct identification signal of the robot. After the IR signature of the robot is verified, the NodeMCU will execute an Embedded C routine to open the solar charging circuit by closing the 5V Relay module. The code includes a stabilized IR signal of 500 ms to prevent electrical arcing and increase battery safety by making the charging terminals act whenever a steady IR signal appears.

To reduce the size of the heap memory footprint and avoid the Watchdog Timer (WDT) resets the code does not use the heavy string manipulation, instead direct integer-to-float conversions of battery voltage and area coverage measurements are used. To balance Wi-Fi power consumption and real-time monitoring, transmission frequency is capped to 1 Hz (one update per second). Analysis of the binary that had been made experimentally demonstrated that memory usage was just 32 percent of the flash memory of the ESP32, with considerable headroom left to future experimentation with more sophisticated SLAM (Simultaneous Localization and Mapping) algorithms.



Fig 12: The charging of robot in dock



Fig 13: The autonomous floor cleaning robot

V. CONCLUSION

The creation of a robot that cleans floors using solar power is a big step forward in combining renewable energy with automation. This project showed that a robot with many functions, powered by a smart solar charging station and controlled through the internet, can be a sustainable and efficient alternative to traditional cleaning methods that rely on the grid. It's a significant breakthrough in making cleaning solutions more environmentally friendly and reducing our reliance on non-renewable energy sources. The robot's ability to operate autonomously and efficiently makes it an attractive option for various industries, and its use of solar power reduces its carbon footprint and operating costs. Overall, this innovation has the potential to revolutionize the way we approach cleaning and maintenance, making it more sustainable, efficient, and cost-effective.

When it comes to the actual design of the machine, using a round shape made a big difference in how well it could move around in tight spaces inside. By combining two types of tiny computers, the ESP32 and NodeMCU, we were able to split up the work in a way that made sense - one handled getting around in real-time, while the other made sure the power was used wisely. What we found out was pretty impressive: the system we used to avoid obstacles, which relied on sound waves, worked correctly about 95% of the time. And when it came to cleaning, the part that could both suck up dirt and mop the floor was able to get the job done 42% faster than if we had done it by hand, one step at a time.

One of the main things this project has achieved is making a solar-harvesting docking system work. By using the strongest sunlight to reduce battery drain, the system can run for longer without needing human help, which is especially useful in places like Madurai that get a lot of sunlight. Also, by using the Blynk IoT platform, the robot can now connect to the internet and send information, rather than just working on its own. The fact that it can send alerts when the battery is low in under 2 seconds and can dock automatically with high precision shows that it's ready to be used in large groups with minimal human supervision. This is a big step forward, as it means the system can work independently for longer periods, making it more efficient and reliable. The ability to deploy it on a large scale could have a significant impact, especially in areas where human supervision is limited.

The robot we've been working on can already do a great job of cleaning on its own, but it does have a bit of trouble getting into tight 90-degree corners because of its round shape. Despite this, we're really happy with how it's turned out - it can move around and clean all by itself, and it's really good at getting the job done. Next, we want to make it even smarter by adding a special system that helps it map out its surroundings and figure out where it is. We're also looking into using a new kind of battery that will let it keep going for even longer. Overall, this project is an important step towards creating a cleaner, more sustainable future for our cities. We think it has a lot of potential to make a big impact, and we're excited to see where it takes us. With a few more tweaks and improvements, we could have a whole fleet of these robots keeping our streets clean and green.



VI. ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the management and faculty members of K.L.N College of Engineering for providing the necessary facilities and support to carry out this project. We would also like to thank our project guide for their continuous guidance and valuable suggestions throughout the development of this system.

REFERENCES

1. Arduino Based Voice Controlled Robot by K. Kannan, Dr. J. Selvakumar PG Scholar, Embedded System Technology, SRM University, Tamil Nadu, India AP (S.G), Tamil Nadu, India: ECE Department, SRM University, International Research Journal of Engineering and Technology (IRJET) 2015.
2. Floor Cleaning Robot (Autonomous/ Manual), Vatsal N Shah, Sathvik K P V S, Indus University, Ahmedabad, India, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-018 INCIMACEMT - 2016 Conference Proceedings.
3. Design and implementation of a cost-effective vacuum cleaner robot by Aml Eren, Hatice Dogan, Izmir, Turkey: Dokuz Eylul University, Engineering Faculty, Electrical and Electronics Engineering Department, Turkish Journal of Engineering - 2022.
4. Arduino Based Voice Controlled Robot, Aditya Chaudhry, Manas Batr, Prakhar Gupta, Sahil Lamba, Suyash Gupta, Information Technology, Bharati Vidyapeeth's College Of Engineering, New Delhi, 2019 International Conference on Computing, Communication, and Intelligent Systems.
5. Implementation Of Goods Lifter Robots With Android Based Voices Orders, Hidyatullah, Jhonson Efendi Hutagalung, Royal Kisaran, Study Programm Of Computer Technic, Kisaran Indonesia, IOP Conf. Series: Journal of Physics: Conf. Series 1255 (2019).
6. Floor cleaning robot with reconfigurable mechanism, Veerajagadheswar Prabakarana, Mohan Rajesh Elaraa, Thejus Pathmakumara, Shunsuke Nansaib, Engineering Product Development, Singapore University of Technology and Design, Automation in Construction 91 (2018).
7. K. Uday and C. S. Aravind, "Automatic sensing and switching the water pump: A new approach", 4th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 170-173, doi: 10.1109/ICICCS48265.2020.9121147, 2020.
8. A. P. Murdan and P. K. Ramkissoon, "A smart autonomous floor cleaner with an Android-based controller", 3rd International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering (ELECOM), pp. 235-239, doi: 10.1109/ELECOM49001.2020.9297006, 2020.
9. Burman, V., Kumar, R. "IoT-Enabled Automatic Floor Cleaning Robot", Int. Conference on Recent Advances in Mechanical Engineering ICRAME 2020. Lecture Notes in Mechanical Engineering. Springer, Singapore. 2021. https://doi.org/10.1007/978-981-15-9678-0_60
10. C.Nagarajan and M.Madheswaran - 'Stability Analysis of Series Parallel Resonant Converter with Fuzzy Logic Controller Using State Space Techniques'- Taylor & Francis, Electric Power Components and Systems, Vol.39 (8), pp.780-793, May 2011. DOI: 10.1080/15325008.2010.541746
11. C.Nagarajan and M.Madheswaran - 'Experimental verification and stability state space analysis of CLL-T Series Parallel Resonant Converter' - Journal of Electrical Engineering, Vol.63 (6), pp.365-372, Dec.2012. DOI: 10.2478/v10187-012-0054-2
12. C.Nagarajan and M.Madheswaran - 'Performance Analysis of LCL-T Resonant Converter with Fuzzy/PID Using State Space Analysis'- Springer, Electrical Engineering, Vol.93 (3), pp.167-178, September 2011. DOI 10.1007/s00202-011-0203-9
13. S.Tamilselvi, R.Prakash, C.Nagarajan, "Solar System Integrated Smart Grid Utilizing Hybrid Coot-Genetic Algorithm Optimized ANN Controller" Iranian Journal Of Science And Technology-Transactions Of Electrical Engineering, DOI10.1007/s40998-025-00917-z,2025
14. S.Tamilselvi, R.Prakash, C.Nagarajan, " Adaptive sliding mode control of multilevel grid-connected inverters using reinforcement learning for enhanced LVRT performance" Electric Power Systems Research 253 (2026) 112428, doi.org/10.1016/j.epr.2025.112428
15. S.Thirunavukkarasu, C. Nagarajan, 2024, "Performance Investigation on OCF and SCF study in BLDC machine using FTANN Controller," Journal of Electrical Engineering And Technology, Volume 20, pages 2675-2688, (2025), doi.org/10.1007/s42835-024-02126-w



16. Nagarajan, M.Madheswaran and D.Ramasubramanian- 'Development of DSP based Robust Control Method for General Resonant Converter Topologies using Transfer Function Model'- Acta Electrotechnica et Informatica Journal , Vol.13 (2), pp.18-31, April-June.2013, DOI: 10.2478/aei-2013-0025.
17. C.Nagarajan and M.Madheswaran - 'DSP Based Fuzzy Controller for Series Parallel Resonant converter'- Springer, Frontiers of Electrical and Electronic Engineering, Vol. 7(4), pp. 438-446, Dec.12. DOI 10.1007/s11460-012-0212-0.
18. C.Nagarajan and M.Madheswaran - 'Experimental Study and steady state stability analysis of CLL-T Series Parallel Resonant Converter with Fuzzy controller using State Space Analysis'- Iranian Journal of Electrical & Electronic Engineering, Vol.8 (3), pp.259-267, September 2012.
19. C.Nagarajan and M.Madheswaran, "Analysis and Simulation of LCL Series Resonant Full Bridge Converter Using PWM Technique with Load Independent Operation" has been presented in ICTES'08, a IEEE / IET International Conference organized by M.G.R.University, Chennai.Vol.no.1, pp.190-195, Dec.2007
20. Suganthi Mullainathan, Ramesh Natarajan, "An SPSS and CNN modelling based quality assessment using ceramic materials and membrane filtration techniques", Revista Materia (Rio J.) Vol. 30, 2025, DOI: <https://doi.org/10.1590/1517-7076-RMAT-2024-0721>
21. M Suganthi, N Ramesh, "Treatment of water using natural zeolite as membrane filter", Journal of Environmental Protection and Ecology, Volume 23, Issue 2, pp: 520-530,2022
22. P. Singh, P. Nayak, A. Datta, D. Sani, G. Raghav and R. Tejpal, "Voice Control Device using Raspberry Pi", Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 2019, pp. 723-728, doi: 10.1109/AICAI.2019.8701409.
23. Yunbo Hong, Rongchuan Sun, Rui Lin, Shumei Yu and Lining Sun, "Mopping module design and experiments of a multifunction floor cleaning robot ", Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, pp. 5097-5102, 2018, doi: 10.1109/WCICA.2014.7053581.
24. H. A. S. H. Prayash, M. R. Shaharear, M. F. Islam, S. Islam, N. Hossain and S. Datta, "Designing and Optimization of An Autonomous Vacuum Floor Cleaning Robot ", IEEE International Conference on Robotics, Automation, Artificial Intelligence and Internet-of-Things (RAAICON), Dhaka, Bangladesh, pp. 25-30, 2019 doi: 10.1109/RAAICON48939.2019.11.
25. V. P.H, L. V, M. K, R. P.S and S. R, "Sweepy-The Smart Floor Cleaner," International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), Bangalore, India, 2018, pp. 124-126, doi: 10.1109/ICDI3C.2018.00035.
26. H. G. T. Milinda and B. G. D. A. Madhusanka, "Mud and dirt separation method for floor cleaning robot ", Moratuwa Engineering Research Conference (MERCon), Moratuwa, Sri Lanka, pp. 316-320, 2019, doi: 10.1109/MERCon.2017.7980502.
27. Yunbo Hong, Rongchuan Sun, Rui Lin, Shumei Yu and Lining Sun, "Mopping module design and experiments of a multifunction floor cleaning robot ", Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, pp. 5097-5102, 2018, doi: 10.1109/WCICA.2014.7053581.
28. G. Vimal Raja, K. K. Sharma (2014). Analysis and Processing of Climatic data using data mining techniques. *Envirogeochimica Acta 1 (8):460-467*.
29. Mathew, A. A Secure, Trustworthy, and Regulated Framework for AI Agents in Distributed Networks.
30. Sugumar, R. (2025). An Intelligent Cloud-Native GenAI Architecture for Project Risk Prediction and Secure Healthcare Fraud Analytics. *International Journal of Research and Applied Innovations, 8(Special Issue 2), 1-7*.