vehicles

MDPI

Article

Real-Time Embedded Intelligent Control of Hybrid Renewable Energy Systems for EV Charging

Khechchab Adam * and Senhaji Saloua

Sciences, Engineering and Sustainable Development (SIDD), Doctoral Studies Center (CEDoc), Private University of Fès (UPF), Lotissement Quaraouiyine Route Ain Chkef, Fès 30000, Morocco; salouasenhaji@gmail.com
* Correspondence: adam.khechchab00@gmail.com; Tel.: +212-70-6018365

Abstract

In response to the challenges of electric mobility in off-grid contexts, this study introduces a novel and pragmatic solution: an intelligent, embedded EV charging system capable of anticipating energy availability using external weather forecasts. An embedded Model Predictive Control (MPC) scheme was implemented on an ESP32 microcontroller, incorporating real-time solar and wind forecasts transmitted via LoRa. Unlike conventional approaches that are often centralized or resource-intensive, the proposed architecture enables localized, forecast-aware decision making, while respecting physical constraints (SOC, power limits, system stability) within the limits of embedded hardware. The proposed system was fully validated through functional simulations (data acquisition, processing, display, and physical actuation). Results confirm the feasibility of real-time, stable, and proactive energy management, laying the foundation for smart, resilient, and autonomous renewable-based EV charging stations tailored to remote areas and decentralized microgrids.

Keywords: off-grid EV charging; solar–wind energy; Model Predictive Control (MPC); ESP32; weather forecasting; LoRa; embedded systems; smart microgrids

check for updates

Academic Editor: Adolfo Dannier

Received: 14 July 2025
Revised: 18 August 2025
Accepted: 26 August 2025
Published: 15 October 2025

Citation: Adam, K.; Saloua, S. Real-Time Embedded Intelligent Control of Hybrid Renewable Energy Systems for EV Charging. *Vehicles* **2025**, *7*, 116. https://doi.org/10.3390/vehicles7040116

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The transition to sustainable mobility requires a rethinking of electric vehicle (EV) charging infrastructures, especially in remote or off-grid areas. In such contexts, energy autonomy relies on local renewable sources—primarily solar and wind power—which are inherently intermittent and difficult to predict. As a result, ensuring reliable charging service while efficiently utilizing available resources represents a major challenge.

Traditional energy management approaches, often based on fixed rules, quickly reach their limits when faced with weather uncertainty and fluctuating load demands. Although artificial intelligence (AI) [1], particularly machine learning, has been explored for energy forecasting, these methods are often ill-suited for low-power embedded systems, such as microcontrollers.

In response to these challenges, we propose an alternative strategy based on Model Predictive Control (MPC) [2], executed in real time on a low-cost ESP32, a low-power 32-bit System-on-Chip (SoC) microcontroller with Wi-Fi and Bluetooth. The MPC anticipates energy needs over several days using weather forecasts transmitted via LoRa (long range, low-power wireless protocol for IoT communications) from a remote meteorological station and dynamically adjusts EV charging power accordingly. This strategy not only helps to optimize energy utilization, but also preserves battery state of charge in anticipation of future shortages [3].

Beyond the technical demonstration, this study investigates the ability of a low-cost embedded system to make intelligent energy decisions under constrained conditions. The objective is to provide a predictable, robust, and autonomous solution, leveraging low-power long-range communication technologies such as LoRa [4].

2. State of the Art Energy Management Strategies for Off-Grid Charging Stations for Electric Vehicles

Intelligent energy management of hybrid renewable energy systems has gained growing attention in recent years, especially for isolated microgrids and electric vehicle (EV) charging stations. Several strategies have been investigated, ranging from rule-based logics to advanced optimization and learning algorithms.

A previous study of Khechchab et al. [5] evaluated the effectiveness of grid-connected, solar-only, and hybrid solar–wind energy systems for EV charging. While that study focused on the energetic performance of each scenario, it did not address the operational control layer, especially under uncertainty and dynamic user demand.

Among optimization-based approaches, Model Predictive Control (MPC) has been widely adopted to anticipate future operating conditions and make constrained decisions over a prediction horizon. Zhao et al. [6] introduced a distributed robust MPC framework for multi-microgrids, demonstrating improvements in system reliability, battery management, and reduced reliance on external energy sources. Similarly, Alarcón et al. [7] applied an economic MPC strategy to grid-connected microgrids, highlighting its ability to optimize operational costs while effectively balancing intermittent renewable resources. However, most of these implementations remain at the simulation stage and do not address embedded integration with real-time data acquisition.

Some recent efforts have attempted real-time embedded implementations of control strategies. For example, Amar et al. [8] implemented a real-time MPC scheme for a hybrid microgrid using embedded components, but without integrating weather forecasting or validating the system over extended prediction horizons. In addition, García-Torres et al. [9] highlighted that many existing approaches still rely on simplified or rule-based control logics, rather than fully predictive optimization. Compared to these efforts, our work proposes a full end-to-end architecture, including long-range forecast acquisition via LoRa, real-time embedded MPC execution, and integrated system feedback—a combination that remains largely unexplored in the current literature.

On the other hand, machine learning (ML) has emerged as a promising tool for forecasting renewable generation and optimizing EV charging patterns. As highlighted in recent reviews [10], ML-based strategies provide high adaptability and can significantly enhance scheduling and energy management. However, such techniques usually require substantial computational resources, sizeable memory, and large datasets, which limit their deployment on low-power embedded hardware such as ESP32.

Furthermore, most prior works rely on either local sensor data or offline forecast inputs, assuming reliable infrastructure and permanent connectivity. Very few studies have explored the integration of external meteorological forecasts via low-power protocols like LoRa in an embedded control loop. This real-time, forecast-based decision making is especially relevant for off-grid, autonomous renewable systems, yet it remains underaddressed in the current literature.

To fill this gap, we implement a forecast-aware MPC algorithm on a low-cost ESP32 microcontroller, using multi-day irradiance and wind speed forecasts obtained from a remote weather station. This anticipatory approach enables dynamic charging power control that accounts for both future resource availability and embedded system limitations.

3. System Architecture for Forecast-Based Embedded EV Charging Control

The system architecture is built around a complete embedded chain designed to autonomously manage electric vehicle charging from hybrid renewable sources. At its core, an ESP32 microcontroller coordinates energy inputs (solar and wind), a battery bank, and a power delivery interface. This setup is integrated with an optimization mechanism that relies on weather forecasts, enabling the system to anticipate fluctuations in renewable generation and dynamically adjust charging power.

While Amar et al. [8] demonstrated a real-time MPC strategy for hybrid microgrids using embedded components, their work did not address forecast integration or communication aspects. In contrast, our implementation focuses on a fully embedded real-time system with physical hardware constraints and communication modules.

The design targets remote or weakly connected environments, where grid stability cannot be assumed, and predictive, efficient use of resources is essential.

Unlike many existing systems that rely solely on real-time sensor data or assume perfect foresight, the proposed design incorporates external weather forecasts transmitted from a remote meteorological station. These multi-day forecasts are processed in real time and used by a Model Predictive Control (MPC) algorithm embedded on the microcontroller. This anticipatory approach allows the controller to adjust charging power dynamically while accounting for future availability of renewable resources.

Previous works have explored predictive control strategies in hybrid renewable energy systems. For instance, recent reviews have outlined various MPC-based approaches for standalone and hybrid microgrids, emphasizing their potential to enhance system performance [10]. Nassereddine et al. [11] proposed a forecast-based MPC model for renewable microgrids and highlighted the importance of integrating predictions into control strategies. However, their work remains at the simulation level, without validation on low-cost embedded hardware.

Furthermore, several reviews [2,9,12] provide a comprehensive overview of MPC methods for hybrid energy systems, confirming the relevance of this control strategy. Nevertheless, they also underline the research gap in studies that implement and test these strategies in real-time embedded applications such as off-grid EV charging systems.

3.1. System Overview

The system consists of the following main components:

- Remote Weather Station: Provides daily forecasts of solar irradiance (*Fi*) and wind speed (*Wi*), transmitted via LoRa to the ESP32 controller.
- LoRa Receiver Module: Receives and decodes forecast data packets, which are parsed by the ESP32.
- ESP32 Microcontroller: Hosts the MPC algorithm. Based on current SOC, weather forecasts, and system constraints, it determines the optimal charging power P_{opt}.
- Hybrid Renewable Sources: Solar panels and a wind turbine supply energy to the system. The predicted output of these sources is derived from the weather data.
- Battery Bank: Stores excess energy and powers the EV when production is insufficient. The controller accounts for discharge limits and minimum SOC thresholds.
- Power Converter/Actuator: Receives the control signal from the ESP32 (e.g., PWM or DAC) and regulates power delivery to the EV.
- LCD Display (optional): Shows key system metrics such as SOC, control output, and weather summaries for real-time monitoring.

Functional Block Diagram of the Proposed System

In Figure 1, the diagram illustrates the main functional blocks of the forecast-aware EV charging system. Forecast data is received via LoRa and processed by the ESP32, which calculates the optimal power based on a predictive model and system constraints. The decision is applied through a power converter to the EV load, while a display provides real-time feedback.

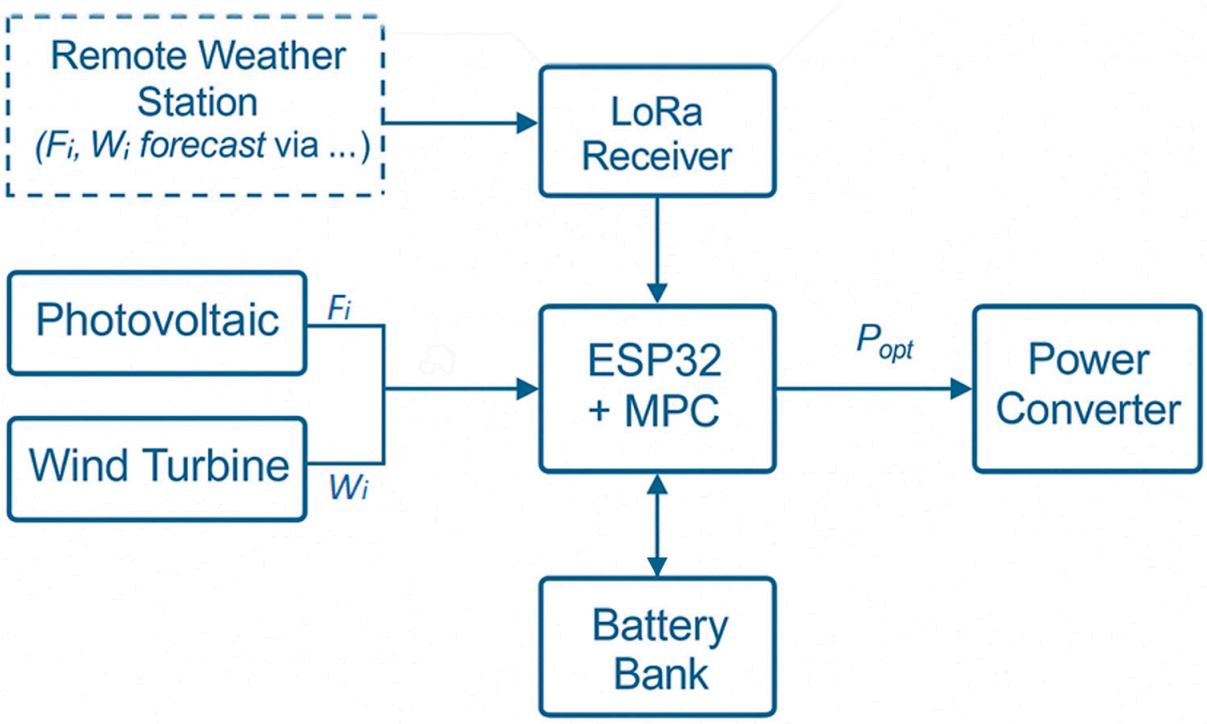


Figure 1. Functional block diagram of the forecast-aware embedded EV charging system.

3.2. Technical Significance

The system is designed to be low-cost, modular, and suitable for deployment in remote or weakly connected environments. As demonstrated in reference [13], LoRa communication is effective for long-range, low-power data transmission in microgrid applications. Building on this principle, our system uses LoRa to transmit forecast data to the embedded controller, enabling timely updates for predictive energy management. Furthermore, the ESP32 microcontroller shows that MPC can be implemented on affordable, resource-limited hardware.

By combining weather-aware control, embedded optimization, and real-time actuation, this architecture contributes a practical, intelligent solution for managing renewable energy in off-grid EV charging stations.

4. Forecast-Aware Embedded MPC for Real-Time EV Charging Control

4.1. Objectives of the Control Strategy

The controller is responsible for ensuring reliable and efficient electric vehicle (EV) charging by managing energy flows between renewable sources, the battery bank, and the charging point. This task becomes particularly challenging in off-grid contexts, where the inherent variability of solar and wind production cannot be mitigated by grid support. To address this uncertainty, the system must be capable of anticipating critical periods and adapting to the available resources while complying with physical and operational constraints.

4.2. Modeling the State of Charge Dynamics

The battery's state of charge (SOC) represents the proportion of stored energy at a given time, relative to the total battery capacity. Its dynamics allow for the prediction of the evolution of the energy level in the battery, according to charging decisions.

It is modeled by the following equation:

$$SOC(t) = SOC(t-1) + \frac{\eta_{charge} \cdot P(t) \cdot \Delta t}{C_{batt}} \quad (1)$$

where

- $SOC(t)$: battery charge level at time t (normalized between 0 and 1);
- $P(t)$: applied charging power (in W);
- η_{charge}: overall charge and conversion efficiency ($0 < \eta \leq 1$);
- Δt: time step duration (in hours);
- C_{batt}: total battery capacity (in Wh).

This equation makes it possible to accurately calculate the battery's charge level at each time step, taking the following into account:

- The electrical power injected into (or drawn from) the system;
- The duration for which this power is applied;
- The potential energy losses due to efficiency.

It plays a central role in MPC simulation logic. At each control cycle, the system predicts the SOC evolution for several candidate power profiles *P[t]* and ensures the following:

- The SOC remains within safety limits (see Section 4.3);
- The profile meets the charging goals (e.g., recharging before a shortage);
- The overall strategy cost is minimal (see Section 4.4).

4.3. Operational Constraints

The system enforces constraints to ensure safe operation.

- The battery SOC must remain within a secure range:

$$SOC_{min} \leq SOC(t) \leq SOC_{max}$$

- The charging power must not exceed the converter or system limits:

$$0 \leq P(t) \leq P_{max}$$

These constraints are accounted for during optimization.

4.4. Integration of the Cost Function into Control Logic

The core of the MPC controller is based on evaluating a cost function that guides the choice of the optimal charging power strategy. This function enables proactive decisions by considering multiple objectives: maintaining a suitable SOC level, ensuring system stability, and efficiently using available renewable energy.

It is formulated as follows:

$$J = \underbrace{\alpha \cdot \left(SOC_{target} - SOC_{fin}\right)^2}_{\text{Target SOC term}} + \underbrace{\beta \cdot \sum_t (P(t) - P(t-1))^2}_{\text{Smooth power variation}} + \underbrace{\gamma \cdot \sum_t \left(P(t) - E_{gen}(t)\right)^2}_{\text{Renewable tracking}} \quad (2)$$

where

- α, β, γ: weighting coefficients;

- SOC_{target}: target SOC at the end of the prediction horizon;
- $P(t)$: power applied at time t (in W);
- $E_{gen}(t)$: forecasted available power at time t (in W);
- SOC_{fin}: simulated SOC at the end of the horizon;
- J: overall cost function value used to evaluate each candidate power profile; it quantifies the trade-off between reaching a desired SOC, ensuring smooth control actions, and aligning power usage with renewable availability.

Each term serves a specific purpose, as follows:

- Target SOC term: ensures that the battery reaches a desired SOC by the end of the horizon, ensuring autonomy for coming days.
- Smooth power variation: reduces sudden changes in charging power between consecutive steps, promoting system stability and protecting components.
- Renewable tracking: aligns energy consumption with available renewable production, helping to reduce waste and maximize self-consumption.

The system uses 7-day weather forecasts, with a 3 to 6 h time step, received from a remote station. These forecasted data allow the MPC to proactively plan charging, especially by preserving SOC during upcoming periods of low renewable availability. This formulation allows the system to dynamically prioritize different objectives, depending on the context (e.g., increasing α when anticipating long production shortages or emphasizing γ when excess solar energy is available). In our simulations, the weighting parameters were empirically selected as $\alpha = 1.0$, $\beta = 0.2$, and $\gamma = 0.8$ to achieve a balanced trade-off between target SOC tracking, control smoothness, and renewable energy utilization. These values remained fixed across all tested scenarios.

4.5. Receding Horizon Strategy and Implementation Algorithm

In this study, the EV energy demand is modeled as a predefined charging requirement, which can be constant or time-dependent. This demand profile is assumed to be known in advance and is used as a constraint when evaluating each candidate power sequence. The MPC controller ensures that the delivered power remains feasible with respect to both the renewable availability and the EV's expected needs.

The MPC controller uses a receding horizon strategy, combined with a simulation-based decision process. At each time step, the system not only reacts to the current state, but also anticipates upcoming conditions over several days, based on weather forecasts received via LoRa.

Specifically, the MPC simulates multiple power profiles over a prediction horizon of up to 7 days, with a time step between 3 and 6 h. These forecasts include solar irradiance *Fi[t]* and wind speed *Wi[t]*, from which the expected renewable energy production *Egen(t)* is estimated.

For each candidate profile, the controller

- Predicts SOC evolution using the dynamic model (Section 4.2);
- Evaluates the cost function (Section 4.4);
- Verifies that all operational constraints are satisfied.

After analyzing all profiles, only the first power value $P(t)$from the best profile is applied. This method enables the system to continuously adapt to mid-term weather trends. The main steps of this algorithm are summarized in the pseudo-code shown in Listing 1.

Listing 1. C++ code snippet implementing the forecast-aware MPC loop on ESP32.

```
    # At each control interval (t):
1.  forecast = receive_forecast_LoRa()            # Get Fi[t], Wi[t] for t → t+N
2.  E_gen = estimate_generation(forecast)         # Compute E_pv, E_wind

3.  best_J = ∞
4.  best_P_seq = []

5.  for each candidate sequence P[t...t+N]:
6.   SOC_sim = simulate_SOC(SOC[t], E_gen, P_seq)
7.   J = compute_cost(SOC_sim, E_gen, P_seq)
8.    if J < best_J:
9.  best_J = J
10. best_P_seq = P_seq

11. apply_power(best_P_seq[0])                    # Apply only the first decision
```

5. Embedded Implementation of the Forecast-Aware Control on ESP32

5.1. Embedded Hardware Architecture

The embedded control system is implemented on an ESP32-WROOM microcontroller. It communicates with a remote weather station via LoRa, manages charging power delivery to the EV, and monitors system variables such as battery SOC and energy production. Figure 2 illustrates the embedded hardware architecture of the control system, while Table 1 details its main components and technical specifications.

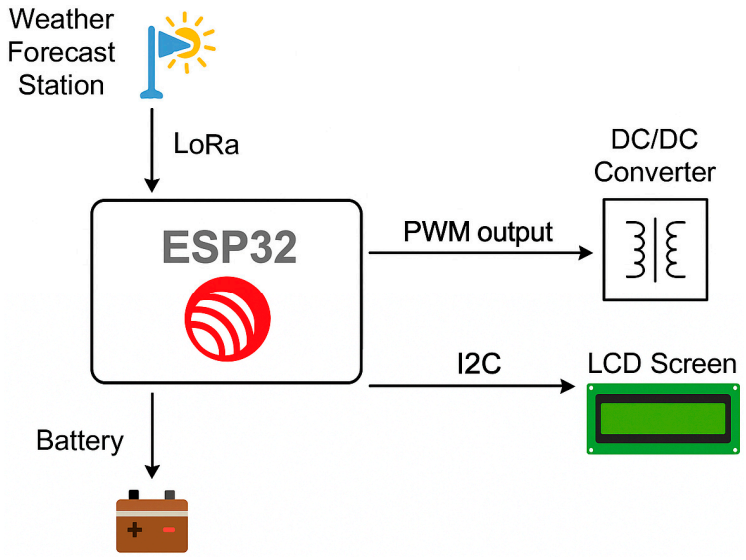


Figure 2. Embedded hardware architecture for real-time MPC-based EV charging control.

Table 1. Technical specifications of the main embedded hardware components.

Component	Model	Function	Key Specifications
Microcontroller	ESP32-WROOM	Control and computation	240 MHz, 520 KB RAM
LoRa Module	SX1278	Receives weather forecasts	433 MHz communication
Display	I2C LCD 16 × 2	User interface	5 V operation
Battery	Li-Ion 12 V	Energy storage	5 kWh capacity
DC/DC Converter	LM2596	Voltage regulation	Input: 12 V; output: 5 V

5.2. Software Environment for Real-Time Control

The control algorithm was implemented using the Arduino programming environment, with C++ as the primary language. The ESP32 microcontroller provides a suitable platform for smart energy applications, especially in low-cost setups. Building on this foundation, our design integrates a real-time Model Predictive Control (MPC) loop, structured around modular functions for data reception, energy estimation, cost evaluation, and power allocation. The software stack includes the following components:

- Arduino IDE—used for firmware development, debugging, and serial monitoring.
- LoRa.h—for receiving forecast data via a LoRa SX1278 module.
- LiquidCrystal_I2C.h—to display system status and SOC values on a 16 × 2 LCD screen.
- Wire.h—for I2C communication between the ESP32 and peripheral devices.
- EEPROM.h (optional)—used for storing historical SOC values or tuning parameters.
- FreeRTOS (Real-Time Operating System built into the ESP32)—for managing real-time task scheduling and separating computation, communication, and display functionalities.

Weather forecast data for solar irradiance (*Fi[t]*) and wind speed (*Wi[t]*) are received periodically in structured JSON format from a remote weather station via LoRa. These values are parsed and stored in the memory to be used in the MPC optimization.

The controller operates with a fixed prediction horizon (typically N = 3 to 5 time steps), making it suitable for embedded execution. Each iteration of the MPC loop involves simulating future battery states, evaluating a set of candidate charging power profiles, and selecting the one that minimizes the cost function. The computational cycle is completed in less than 100 ms, using less than 50% of the available RAM, confirming the feasibility of the approach using low-power hardware. This confirms its suitability for low-power predictive energy management tasks.

5.3. Forecast Reception via LoRa

The forecast system is designed to receive real-time meteorological data from a remote weather station via LoRa communication. The transmitted data includes multi-hour or multi-day forecasts for both solar irradiance (Fi[t]) and wind speed (Wi[t]), encoded in a structured JSON format.

The weather station, equipped with environmental sensors or connected to an online prediction source [14], packages the forecast into JSON objects. Each transmission includes an array of irradiance and wind speed values for the next N time steps. A sample JSON payload is provided as follows in Listing 2:

Listing 2. Example of JSON payload structure transmitted from the remote weather station via LoRa.

```
{
  "irradiance": [320, 280, 150],
  "wind_speed": [4.2, 3.9, 2.5]
}
```

On the ESP32 side, a LoRa SX1278 module is configured to operate in receive mode, with a dedicated FreeRTOS task or interrupt service routine (ISR) managing incoming messages. Upon reception, the forecast string is parsed using the ArduinoJson library, and the extracted values are stored in local memory buffers to be used in the MPC computation. The forecast data is transmitted at low frequency (e.g., every 3 to 6 h) and contains only

two main variables (irradiance and wind speed). The resulting payload remains compact (<50 bytes), well within LoRa's transmission capacity.

To ensure robustness, the system includes simple data validation (checking array lengths, numeric bounds, and timestamps) before applying the data. If no valid forecast is received, the system falls back to default average values or reuses the last known valid forecast.

This architecture enables asynchronous and low-power forecast acquisition over long distances and decouples the energy management logic from the sensor infrastructure. It also allows the controller to anticipate future availability of renewable energy and optimize power delivery accordingly.

5.4. Embedded MPC Computation and Optimization Loop

The core of the embedded control algorithm is based on a simplified Model Predictive Control (MPC) strategy. This approach enables the controller to anticipate future energy availability and make optimal decisions regarding power delivery to the electric vehicle (EV), while ensuring battery constraints and system stability.

At each control interval (e.g., every minute), the ESP32 performs the following steps:

1. Forecast retrieval: It obtains the most recent solar irradiance (*Fi[t]*) and wind speed (*Wi[t]*) values, received in JSON format.
2. Energy generation estimation: Using Equations (1) and (2) from Section 4, the controller computes the predicted energy generation ($E_{gen}(t)$) for the next n time steps.
3. Battery SOC simulation: For each candidate charging power profile $P(t \ldots t + N)$, the controller simulates the battery's state of charge (SOC) using the dynamic model given in Equation (1).
4. Cost function evaluation: Each candidate sequence is evaluated using the cost function J defined in Equation (2), which accounts for deviation from the target SOC, changes in power level, and mismatch between generation and demand.
5. Optimal decision: The controller selects the candidate profile with the lowest cost and applies only the first power value $P(t)$ to the system.

The number of candidate profiles and prediction steps (N) are selected to match the computational capacity of the ESP32. In practice, 10–15 candidate sequences and a horizon of N = 3 steps provide a good tradeoff between performance and speed. The algorithm is implemented in C++, and each MPC cycle is completed in under 100 ms.

This process is executed within a dedicated FreeRTOS task, allowing the controller to remain responsive and avoid blocking communication or display tasks. The algorithm also includes fallback mechanisms in case of forecast loss by reusing previous forecasts or applying a default safe power level.

The modularity of the algorithm makes it easily extendable to more complex control structures or longer prediction horizons and confirms the feasibility of real-time MPC implementation on low-power embedded platforms like the ESP32.

5.5. Real-Time Power Output and System Feedback

To translate the computed optimal power into a physical signal, the system uses a pulse-width modulation (PWM) output on the ESP32. In the current implementation, a standard LED connected to GPIO Pin 4 serves as a visual indicator of the applied charging power level.

The LED brightness is modulated in real-time, proportional to the power command calculated by the MPC. This simple yet effective feedback mechanism allows visual verification of the control algorithm's response to forecast variation. In parallel, a 16 × 2 I2C LCD displays the system's key variables, including the battery state of charge (SOC), the se-

lected power, and the forecasted energy production. This ensures local monitoring without requiring a computer or serial terminal.

Figure 3 shows a snapshot of the Wokwi simulation, illustrating the ESP32 board, the LED output, and the LCD display updating in real-time during system operation.

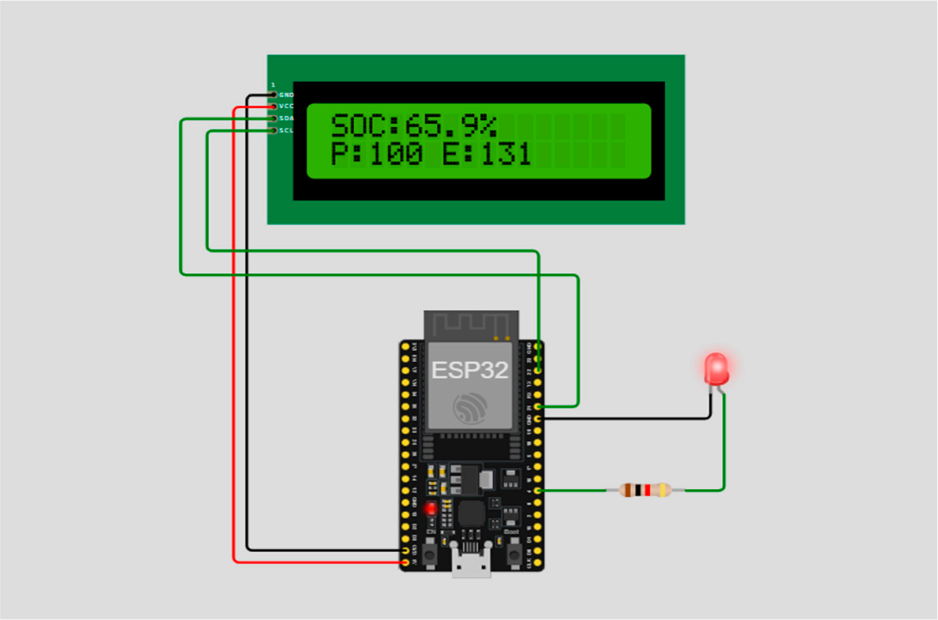


Figure 3. Real-time embedded output display on ESP32 using Wokwi.

This real-time output functionality validates the embedded system's ability not only to compute the control logic, but also to execute the corresponding action within a practical interface. It confirms the tight coupling between algorithmic computation and physical actuation on a low-cost platform.

5.6. Simulation Results of the Embedded MPC Controller

The simulation results presented in this section were generated using the Wokwi platform, which emulates an ESP32 system in real-time, with LCD output and PWM control. Two scenarios were tested:

- A short-term scenario (Figure 4) using a controlled sinusoidal energy generation pattern;

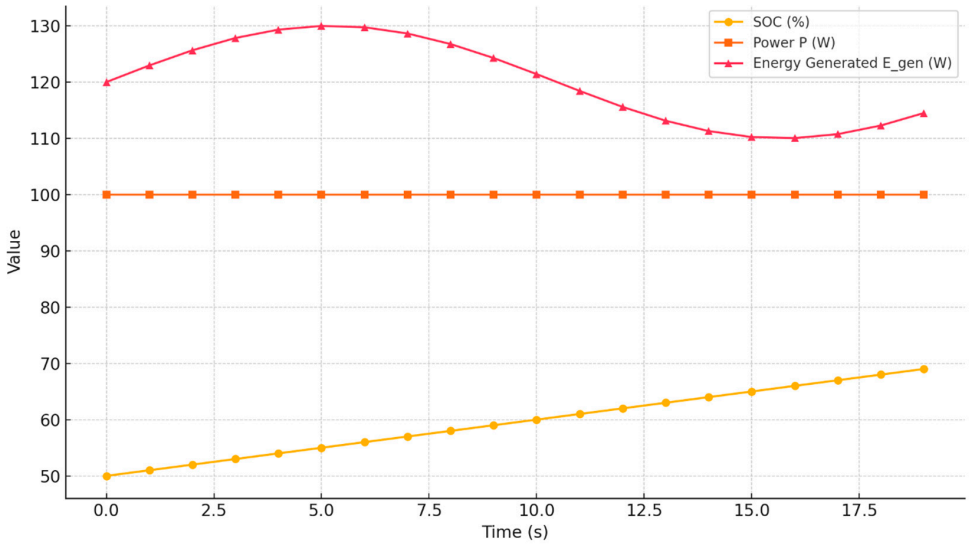


Figure 4. Short-term simulation results based on the embedded controller.

- A long-term scenario (Figure 5) over 30 days, based on synthetic solar and wind forecasts with periodic trends and random noise.

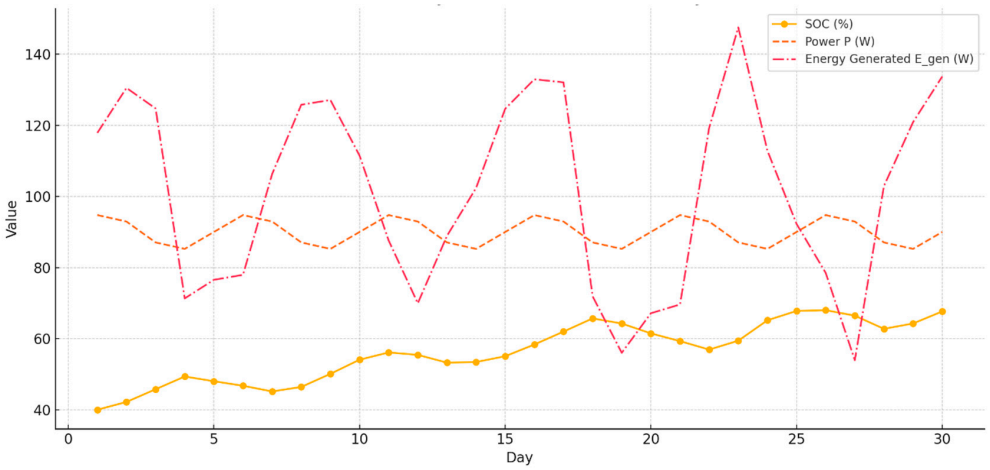


Figure 5. Long-term simulation of SOC, power, and energy over 30 days.

5.6.1. Short-Term Simulation Analysis

Figure 4 shows the evolution of the SOC, the applied charging power, and the simulated available energy over several control cycles.

The following can be observed:

- The SOC increases steadily, showing that the selected power profiles align with available energy;
- While generation fluctuates sinusoidally, the controller maintains a relatively smooth and stable charging power;
- No instabilities or overshoots occur.

This confirms the MPC's ability to filter out production noise and ensure a stable response in dynamic conditions.

5.6.2. Long-Term Simulation Analysis

Figure 5 presents the evolution of the SOC, charging power, and forecasted energy over 30 days.

Key observations:

- The SOC always remains within the safe operational range, respecting all constraints (no deep discharge or overcharging);
- During days 6–7 and 21–23, the SOC plateaus or slightly decreases, indicating that the controller anticipates energy shortages and adjusts its decisions accordingly;
- The SOC never exceeds 70%, despite surplus energy, suggesting that the MPC intentionally avoids overcharging, likely to protect battery health.

These results demonstrate that the embedded MPC responds proactively, autonomously, and robustly to environmental variability. In summary, these simulations validate the following:

- The ESP32-based MPC operates in real time;
- It makes smart and anticipative decisions with limited computing resources;
- It balances optimal charging, system safety, and battery preservation.

6. Hybridization of Solar and Wind Energy for Constant Energy Availability

This section compares Model Predictive Control (MPC) with artificial intelligence (AI)-based approaches for energy management in embedded systems.

6.1. Overview of AI-Based Control Approaches

AI-based control includes machine learning models such as neural networks, support vector machines, and reinforcement learning agents. These models are typically trained offline on large datasets and can capture nonlinear patterns in energy consumption, generation, and user behavior.

6.2. Embedded Implementation Feasibility

While AI algorithms have shown high performance in simulation environments, their integration into low-power embedded systems like ESP32 remains challenging. AI models require more memory and processing power, and inference may induce non-negligible latency, which is problematic for real-time energy management. In contrast, MPC offers lightweight computations, a transparent control structure, and better suitability for resource-constrained platforms.

6.3. Strengths and Weaknesses

Another important aspect is energy consumption at the controller level. MPC, due to its lightweight computational nature and deterministic loop execution, generally consumes less processing time and memory compared to the results for AI models [1]. AI-based approaches, particularly those relying on neural networks, require more frequent memory access, floating-point operations, and occasionally dedicated accelerators—making them less energy efficient on platforms like ESP32. In real-time embedded contexts, where both energy autonomy and low thermal footprint are critical, this makes MPC a more suitable choice [1,2].

A comparative summary of MPC and AI/ML-based control approaches is provided in Table 2, highlighting their relative performance in terms of transparency, real-time suitability, and embedded feasibility.

Table 2. Comparative analysis of MPC and AI-Based control techniques for embedded energy management. This table summarizes key differences between Model Predictive Control and AI-based approaches in the context of embedded energy management systems. Criteria include control transparency, computational complexity, data dependency, and suitability for real-time execution on microcontrollers like ESP32.

Criteria	MPC	AI/ML-Based Control
Transparency	High (equation-based)	Low (black-box model)
Real-time performance	Fast and deterministic	May be slower on microcontrollers
Data dependency	Low (model-based)	High (requires large datasets)
Adaptability	Moderate (fixed model structure)	High (if retrainable online)
Embedded suitability	Excellent for ESP32	Limited unless heavily optimized

6.4. Justification for Choosing MPC

Given the limited computational resources of the ESP32 and the need for deterministic and interpretable control, MPC was selected as the most appropriate method. It allows anticipatory control using forecasts without requiring large datasets or model training, and its performance was validated in both short-term and long-term scenarios.

Moreover, MPC ensures consistent execution timing and a minimal memory footprint, which are critical in real-time embedded environments. Compared to AI approaches that require model inference and possibly floating-point operations, MPC remains faster and more energy-efficient [1,2].

While AI techniques offer higher adaptability and potential learning from complex patterns, they introduce additional complexity in embedded implementation and valida-

tion. Therefore, the use of MPC provides an optimal compromise between performance, explainability, and embedded feasibility.

7. Conclusions

In a context where intelligent energy management for electric vehicle (EV) charging remains a challenge—particularly in off-grid environments—this study offers an original contribution by combining weather forecasting, predictive control, and a low-cost embedded architecture.

Unlike existing works that rely on either centralized or resource-intensive approaches, the proposed approach demonstrates the feasibility and effectiveness of embedded MPC control on an ESP32 microcontroller, integrating multi-source weather forecasts received in real time via LoRa. This ability to anticipate, at low cost and with limited resources, provides a practical and applicable solution, especially for remote areas or decentralized infrastructures.

The proposed approach stands out due to the following characteristics:

- The actual and optimized implementation of an embedded MPC algorithm, compatible with the memory and real-time constraints of a microcontroller;
- The explicit integration of external weather forecasts into the embedded decision-making process—an aspect still rarely addressed in the literature;
- A complete end-to-end simulation, including LoRa data acquisition, energy estimation, real-time visualization, and validation through dynamic testing.

The results demonstrate stable, proactive, and robust energy management, with the SOC maintained within optimal bounds, even under high variability of renewable resources. While this study focuses on simulation-based validation, a physical prototype of the embedded controller has also been developed and tested in follow-up research. These experimental results will be presented in a separate study.

Beyond this technical demonstration, the work invites reflection on new models of distributed energy autonomy. The proposed approach could evolve toward systems capable of learning from actual usage, self-adapting to local contexts, or even collaborating with other stations to form intelligent and interconnected clusters [4].

This work lays the foundations for a local, anticipatory, and decentralized control architecture, contributing to the emergence of a new generation of embedded solutions—combining technological simplicity, energy resilience, and embedded intelligence

Author Contributions: K.A. undertook the literature review and main writing of the paper. K.A. and S.S. developed the control structure and processed the results. K.A. and S.S. reviewed and edited the whole document, including the figures. S.S. guided the research work. All authors have read and agreed to the published version of the manuscript.

Funding: This research did not receive any specific support from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

References

1. Preuveneers, D.; Joosen, W. Resource Usage and Performance Trade-offs for Machine Learning Models in Smart Environments. *Sensors* **2020**, *20*, 1176. [CrossRef]
2. Hu, J.; Karimi, H.; Huang, B.; Sun, Y. Model Predictive Control of Microgrids—An Overview. *Renew. Sustain. Energy Rev.* **2021**, *136*, 110422. [CrossRef]

3. Negri, G.; Bianchini, A.; Ferrara, C.; Gamberi, M.; Graziani, A.; Manzini, R. Combined Model Predictive Control and ANN-Based Forecasters for Jointly Acting Renewable Self-Consumers: Environmental and Economic Evaluation. *Renew. Energy* **2022**, *199*, 1208–1223. [CrossRef]
4. Pasetti, M.; Zanella, A.; Zorzi, M. On the Use of LoRaWAN for Energy-Efficient IoT Applications in Smart Cities. *Appl. Sci.* **2020**, *10*, 320. [CrossRef]
5. Khechchab, A.; Senhaji, S. Evaluating Charging Systems for Electric Vehicles: Grid vs. Solar Power and the Role of Solar-Wind Hybrid Solutions. *Sci. Afr.* **2025**, *29*, e02793. [CrossRef]
6. Zhao, Z.; Xu, Y.; Zhang, W.; Huang, S. Distributed Robust Model Predictive Control-Based Energy Management Strategy for Islanded Multi-Microgrids Considering Uncertainty. *IEEE Trans. Smart Grid* **2022**, *13*, 2107–2120. [CrossRef]
7. Alarcón, M.A.; Alarcón, R.G.; González, A.H.; Ferramosca, A. Economic Model Predictive Control for Energy Management of a Microgrid Connected to the Main Electrical Grid. *J. Process Control* **2022**, *117*, 40–51. [CrossRef]
8. Amar, A.; Yusupov, Z. Real-Time Capable MPC-Based Energy Management of Hybrid Microgrid. *Processes* **2025**, *13*, 2883. [CrossRef]
9. García-Torres, F.; Bordons, C.; Roncero-Sánchez, P. Model Predictive Control for Microgrid Functionalities: Review and Future Challenges. *Energies* **2021**, *14*, 1296. [CrossRef]
10. Allwyn, R.G.; Al-Hinai, A.; Margaret, V. A Comprehensive Review on Energy Management Strategy of Microgrids. *Energy Rep.* **2023**, *9*, 5565–5591. [CrossRef]
11. Nassereddine, K.; Saidi, L.; Belkacem, S.; Chellali, B. Simulation of Energy Management System Using Model Predictive Control in a Microgrid. *Sci. Rep.* **2025**, *15*, 14567. [CrossRef] [PubMed]
12. Babić, L.; Stanković, A.M.; Đurović, Ž. Data-Driven Non-Parametric Chance-Constrained Model Predictive Control for Microgrid Energy Management. *Front. Control Eng.* **2023**, *4*, 1237759. [CrossRef]
13. Ndukwe, C.; Iqbal, M.T.; Liang, X.; Khan, J.; Aghenta, L. LoRa-Based Communication System for Data Transfer in Microgrids. *AIMS Electron. Electr. Eng.* **2020**, *4*, 303–325. [CrossRef]
14. NASA Prediction Of Worldwide Energy Resources (POWER) Project. POWER Data Access Viewer. Available online: https://power.larc.nasa.gov/data-access-viewer/ (accessed on 10 September 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.