Sustainably Powered System Designed to Transform Food Production and Foster Self-Sustaining Societies with Artificial Intelligence

This study focuses on the development of a robust system engineering to support sustainable agricultural practices. It emphasizes the integration of IoT technologies, such as soil-embedded sensors, with clean energy sources to create self-sufficient and efficient agricultural systems. Artificial intelligence is introduced as a complementary tool, leveraging the engineered system to process data and support decision-making, further improving efficiency and sustainability.

A study by Victor Sales

For Faculdade Pitágoras / University of South Bohemia

Contagem, 2025.

SUMMARY

INTRODUCTION

1. What this study will address

While the title of this study hints at a broad scope, the primary focus lies in system engineering, specifically the integration of key technologies for agricultural solutions. This includes the Internet of Things (IoT) through the use of microcontrollers like Arduino Leonardo and ESP32, along with sensors for data collection; and Data Management, encompassing acquisition, storage, and display through relational databases, backend development, and user-facing frontend interfaces.

Adjacent aspects, such as the energy supply for microcontrollers, potential real-world applications of these systems, and their societal benefits, are briefly addressed to provide context and highlight the broader impact of the engineered solutions. Artificial Intelligence (e.g., Machine Learning and Computer Vision) is introduced in a supportive role, leveraging the system's architecture to further enhance decision-making and efficiency.

2. What this study will not address

While energy efficiency is essential for the functionality and stability of the system proposed in this research, this study will not explore the technical specifics of energy management or delve into areas that lie outside the core domain of Computer Science.

Additionally, while Artificial Intelligence (AI) is referenced as a tool to enhance decision-making and system efficiency, this study will not delve into the specifics of AI model development or advanced machine learning techniques, as they are not the central focus of this research.

Similarly, although the study briefly touches upon a potential social model for implementing the system, it does not conduct an in-depth analysis of the socio-economic impacts or broader societal implications of its adoption. These aspects are considered beyond the scope of this research.

3. The Social Problem

According to the United Nations, 3.1 billion people worldwide live in food insecurity, meaning they lack assurance of having food within the next 24 hours, or don't have access to nutritious food [1]. This figure becomes even more striking when compared to the percentage of the global population without internet access: 2.7 billion people. [2] How can internet access be more widespread than consistent access to quality food?

These numbers inspired me to propose a solution: a collaborative farming system—small-scale farms capable of supporting communities of a few hundred residents, providing quality food for all inhabitants while minimizing the political, social, or economic factors that often disrupt food distribution in global markets [3].

4. The Technical Challenge

The system to be developed must integrate three key areas of technology:

1. **Hardware Engineering**, to gather real-time soil data through sensors for humidity, temperature, and pressure [3].
2. **Software Engineering**, to receive, store, and make the processed data (via AI) accessible to the end user [4].
3. **Artificial Intelligence**, to analyze raw data and assist the producer in decision-making [5].

Developing this application entails connecting these three layers and delivering actionable results to the end user.

METHODOLOGY

1. System Architecture

As previously introduced, the system operates across three layers, each responsible for a distinct part of the process, and all equally crucial to the technical success of the solution [3][4]. Let us delve into the details and specifications of each of these layers.

- **Physical Layer**

  This layer is subdivided into microcontrollers, sensors, and power supply.
  Starting with the sensors, two devices were utilized: the LM393 for measuring soil moisture and the DS18B20 for measuring ambient temperature.
  Regarding microcontrollers, the Leonardo R3 was employed to connect all components and manage the energy consumption and voltage required for each module. Additionally, the ESP32 microcontroller, connected to the Arduino Leonardo, was used to send sensor data to a predefined network route via Wi-Fi.
  Lastly, a small 5V solar panel was used to capture energy and transfer it to a Li-ion 18650 battery, which stored the energy and supplied it to the Arduino Leonardo as needed.

- **Logical Layer**

  Once data was reliably transmitted from the physical layer, the logical layer was developed. This layer is responsible for receiving, storing, and making processed data accessible to the user. It can be divided into two components: the back-end and the front-end.

  In the back-end, the Java programming language, supported by the Spring framework, was chosen to manage the entire data flow on the server-side. The database selected to store the information—both raw data received from the physical layer and processed data from the AI—was MySQL [3].

  In the front-end, user experience guided the development process. JavaScript was used alongside the React library to create an intuitive and responsive interface for end users.

- **Artificial Intelligence Layer**

  The Artificial Intelligence (AI) layer is a theoretical component, studied as a means to process raw data from the physical layer and generate actionable insights. It explores the potential of techniques like machine learning and computer vision to optimize food production and assist farmers in decision-making [4].

  The AI layer could analyze sensor data, such as soil moisture and temperature, using machine learning models to identify patterns and predict irrigation needs, planting schedules, or risks like pests and diseases [5]. These predictions demonstrate how AI could enhance agricultural practices.

  Studies suggest AI could offer real-time recommendations to farmers, such as when to irrigate or how to optimize soil nutrients. Although not implemented, this theoretical capability illustrates how AI might improve efficiency and resource management [6].

  While not realized yet, this layer highlights the potential for AI to transform agriculture into a data-driven model, improving yield, sustainability, and food security through further research and development.

2. Tools and Techniques

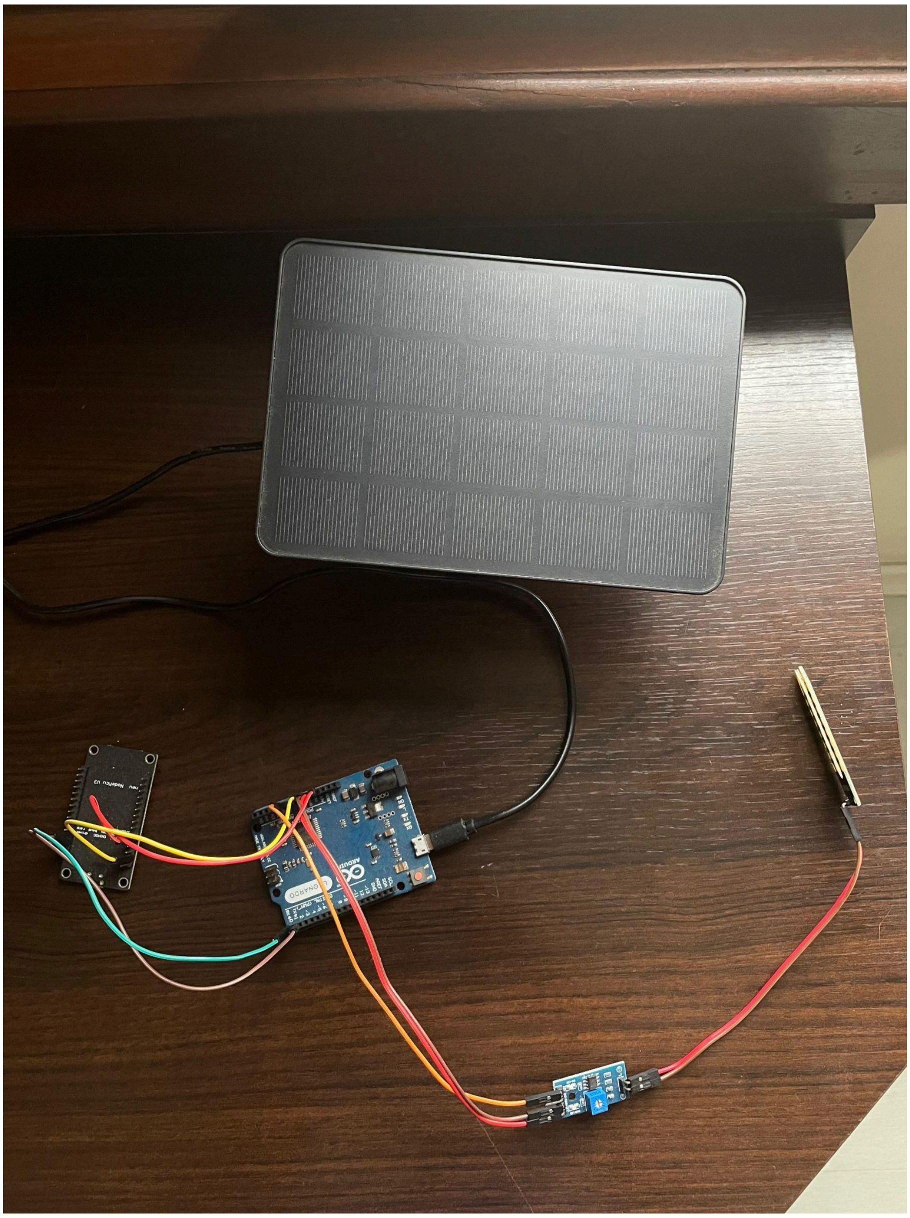
As the physical and logical layers were developed, here is a detailed explanation of the components, frameworks, languages, and technologies used to implement the system.

- Physical Layer

Image 1: Initial soil moisture data collection from a plant.

Initially, the Arduino Leonardo was powered by a computer, and soil moisture data was sent every 10 minutes to the Arduino IDE and then saved in a local MySQL database.

Image 2: ESP32 and Solar Panel integrated with Arduino Leonardo.



At this point, the solar energy captured was already sufficient to power the Arduino Leonardo, the ESP32, and the sensors throughout the day and afternoon. As night fell and solar energy became scarce, the power supply was typically interrupted about 3 to 4 hours after sunset.

The data sent via the internet by the ESP32 was already reaching an endpoint configured in the backend, responsible for processing the data.

- Logical Layer

In the backend, the controller responsible for the initial handling of the data was developed. Its sole responsibilities were to receive the data, forward it to the service that manages the logic, and return any potential errors.

Image 3: Backend controller responsible for receiving data from the physical layer.

```
package com.demeter.controllers;

import com.demeter.services.ArduinoService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.time.LocalDateTime;

@RestController
@RequestMapping("/api")
public class ArduinoController {

    @Autowired
    private ArduinoService arduinoService;

    @PostMapping("/postArduinoData")
    public ResponseEntity<String> postArduinoData( @RequestParam int arduinoId, @RequestParam double
humidity, @RequestParam double temperature, @RequestParam LocalDateTime timestamp) {

        boolean isDataValidAndWasSavedInTheDatabase = arduinoService.validateArduinoData(arduinoId,
humidity, temperature, timestamp);

        if (isDataValidAndWasSavedInTheDatabase) {
            return ResponseEntity.status(HttpStatus.CREATED)
                    .body("Data saved successfully.");
        } else {
            return ResponseEntity.status(HttpStatus.BAD_REQUEST)
                    .body("Invalid data.");
        }
    }
}
```

After the data is received, it needs to be processed before being saved in the database. For this purpose, the service layer is responsible for both processing the data and saving the information to the database.:

Image 4: Service that processes the information and saves it to the database.

```java
package com.demeter.services;

import com.demeter.entities.ArduinoData;
import com.demeter.repositories.ArduinoDataRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.time.LocalDateTime;

@Service
public class ArduinoService {

    @Autowired
    private ArduinoDataRepository arduinoDataRepository;

    public ArduinoData validateAndSaveArduinoData(int arduinoId, double humidity, double temperature,
LocalDateTime timeStamp) {
        if (arduinoId <= 0) {
            throw new IllegalArgumentException("The Arduino ID must be greater than 0.");
        }

        if (humidity < 0 || humidity > 100) {
            throw new IllegalArgumentException("Humidity must be between 0 and 100.");
        }

        if (temperature < -50 || temperature > 100) {
            throw new IllegalArgumentException("Temperature must be between -50 and 100 degrees.");
        }

        if (timeStamp == null) {
            throw new IllegalArgumentException("Timestamp cannot be null.");
        }

        ArduinoData arduinoData = new ArduinoData(arduinoId, humidity, temperature, timeStamp);
        return arduinoDataRepository.save(arduinoData);
    }
}
```

The final part of this layer consists of presenting the data to the user. To achieve this, endpoints were created to consume the data already analyzed by the AI and to provide this data to the frontend.

The user-facing interface is as follows:

Image 5: Dashboard displaying Arduino data with information processed by the AI.

- Artificial Intelligence Layer

  This layer has not yet been implemented; however, a theoretical framework is outlined based on recent advancements in artificial intelligence for agriculture [7].

  The data processed in the logical layer—soil moisture, ambient temperature, crop type, and planting schedules—would serve as inputs for advanced AI algorithms. Using machine learning techniques, such as Recurrent Neural Networks (RNNs) for time-series analysis and Convolutional Neural Networks (CNNs) for image-based crop health assessment, the system could provide precise, actionable insights.

For example:

Irrigation Management: Predictive models, leveraging data like soil moisture and weather forecasts, could determine optimal irrigation schedules, minimizing water usage and improving efficiency.
Harvest Optimization: Algorithms could estimate the best harvest time by correlating crop-specific growth patterns with environmental conditions, enhancing planning and reducing post-harvest losses.
Anomaly Detection: AI could identify anomalies, such as abrupt temperature shifts or irregular moisture levels, signaling potential threats like system failures, pests, or climatic events.
Pest and Disease Prediction: Through pattern recognition in environmental and historical data, models could provide early warnings for pest or disease outbreaks, allowing for proactive intervention.
Resource Optimization: Adaptive learning systems could offer real-time recommendations for soil nutrient adjustments and fertilizer applications based on plant requirements and soil conditions.
Future iterations of the AI layer could integrate generative models, such as Transformers, to provide context-aware recommendations, or reinforcement learning for dynamic decision-making in highly variable conditions.

The integration of these technologies would necessitate a robust computational architecture capable of handling large datasets and ensuring low-latency predictions. Cloud-based solutions, coupled with edge computing for on-site processing, are recommended to balance scalability and responsiveness in remote or underserved areas.

While still theoretical, this AI layer represents a significant step toward data-driven agriculture, with the potential to automate decision-making processes and foster sustainable practices. Future research should prioritize model validation with real-world datasets, as well as the ethical implications of deploying AI in resource-constrained farming communities.

3. Social Model Integration

- Empowering Local Food Production

  The proposed system encourages communities to transition away from reliance on centralized food markets by adopting localized food production practices. IoT and AI technologies empower small-scale farms to meet local nutritional needs effectively [3][7]. Real-time data on soil conditions, crop health, and weather patterns enables precise decision-making, optimizing resources and minimizing waste. This approach aligns with the principles of Community Supported Agriculture (CSA), where localized efforts directly connect producers and consumers, fostering mutual benefit and sustainability [8].

- Fostering Community Collaboration

  A key element of the model is its focus on collective action. Community members share responsibilities for maintaining farms, managing resources, and overseeing production. Transparent data access through user-friendly dashboards ensures informed decision-making and builds trust among participants [3]. The CSA model further reinforces this collaborative approach by promoting shared ownership of agricultural outcomes and cultivating stronger community bonds [8]. This shared governance structure enhances accountability and fosters a sense of mutual responsibility.

- Achieving Economic Independence

  Localized production and distribution reduce dependence on volatile global markets and external suppliers. Any surplus production can be redistributed within the community or sold locally, creating a self-sustaining economic model. By retaining financial resources within the community, this approach fosters economic resilience and provides opportunities for reinvestment in local infrastructure and agriculture [8].

- Building Resilient Societies

  Integrating technology with a collaborative social framework transforms vulnerable communities into self-sufficient, resilient societies. Local, sustainable food production ensures long-term stability and reduces dependence on external factors that often disrupt food security [6][8]. This integration of IoT-driven efficiency and community-based agriculture offers a significant step toward equitable and sustainable agricultural ecosystems capable of addressing food insecurity effectively.

4. Uncertainty and Areas for Improvement

The implementation of the current system revealed specific limitations and opportunities for improvement. These challenges highlight areas where further research and development could enhance the system's performance and reliability.

- Solar Energy Supply
  One significant limitation was the solar panel's inability to power the Arduino continuously for 24 hours. Under optimal conditions, the system operated for approximately 14–18 hours daily, falling short during nighttime and extended periods of limited sunlight. In overcast or rainy weather, operational time dropped to less than 12 hours. To address this, future research could explore integrating more efficient solar panels or hybrid energy systems to extend operational hours [6].

- Energy Optimization and Connectivity
  Addressing the limitations of the current system requires advancements in both energy optimization and connectivity. To enhance energy efficiency, future research could focus on developing methods to make the Arduino more power-efficient, ensuring longer operational hours without additional energy demand [6]. Similarly, the integration of advanced solar panels or improved energy storage solutions, such as hybrid systems or high-capacity battery technologies, could significantly extend the system's functionality under varying weather conditions [8].

  In terms of connectivity, the reliance on a stable Wi-Fi network via the ESP32 microcontroller proved to be a critical vulnerability. Interruptions in internet connectivity compromised the system's reliability, especially in remote or underserved areas. Implementing satellite internet as an alternative could ensure uninterrupted communication between the microcontroller and the network, improving overall system resilience and expanding its applicability to regions with limited infrastructure [3][6].

- Weather-Based Performance Monitoring
  The system's performance was highly influenced by weather conditions. Integrating weather forecasting tools, such as the OpenWeather API, could provide predictive insights into daily operational hours based on climate conditions. This data could be organized into a table correlating weather patterns with expected system uptime, enabling proactive adjustments to the system's operation and energy management [6].

- Future Directions
  Addressing these limitations holds the potential to significantly improve the system's robustness and adaptability. Implementing energy-efficient designs and exploring alternative connectivity options, such as satellite internet, could ensure consistent functionality even in remote or underserved areas [6]. Additionally, integrating predictive weather-based analytics, supported by tools like the OpenWeather API, would enable proactive adjustments to system operations based on climatic conditions, further enhancing reliability [8]. These advancements not only improve the practical applicability of the system in regions with variable weather conditions but also contribute to the broader goal of establishing a sustainable and reliable agricultural support framework [3].

CONCLUSION

1. Main Results

The study successfully designed and implemented a functional system architecture to support sustainable agricultural practices, integrating key technologies in IoT, data management, and clean energy. The primary outcomes are summarized below:

- Physical Layer
  The system demonstrated effective data collection using sensors (LM393 and DS18B20) for soil moisture and temperature, integrated with microcontrollers (Arduino Leonardo and ESP32).
  A solar-powered energy system, while operational for most daylight hours (14–18 hours), revealed limitations during cloudy or rainy conditions, impacting consistent functionality [6].

- Logical Layer
  The backend was implemented using Java and the Spring framework, ensuring seamless data flow and storage in a MySQL database. This layer processed incoming data and prepared it for analysis and visualization.
  The frontend, built with React, presented data through an intuitive dashboard, enabling users to access real-time information about soil conditions and system performance [3][4].

- Theoretical Exploration of AI Layer
  Although not implemented, the study outlined how machine learning could enhance decision-making [7]. Theoretical models proposed included:
  Predictive analysis for irrigation and harvest scheduling.
  Real-time anomaly detection in environmental data.
  Adaptive recommendations for optimizing soil nutrients and water usage [5].

- System Limitations
  The system highlighted challenges such as solar power dependency and connectivity issues with the ESP32 microcontroller. These findings provided valuable insights into areas requiring further research, such as energy optimization and alternative connectivity solutions [6].

Overall, the study demonstrated the feasibility of integrating IoT and system engineering for sustainable agriculture while identifying key areas for future development.

2. Conclusion

This research underscores the transformative potential of integrating IoT, clean energy, and artificial intelligence into agricultural systems. By designing and testing a prototype, the study validated the ability to collect, process, and present data effectively to support farming decisions. However, the findings also revealed technical limitations, such as inconsistent solar energy supply and reliance on stable internet connectivity, emphasizing the need for continued innovation [6][8].

The theoretical exploration of the AI layer highlighted its potential to revolutionize agricultural practices through data-driven insights, predictive analytics, and resource optimization. While these capabilities were not implemented, they represent a promising direction for future research [5].

Addressing the identified limitations—such as improving energy efficiency, incorporating satellite connectivity, and utilizing weather-based performance monitoring—will be critical for enhancing system reliability and scalability. These advancements could pave the way for a fully automated and sustainable agricultural solution, capable of improving food production and supporting self-sustaining communities globally [3][8].

This study contributes to the broader field of sustainable agriculture by demonstrating the viability of a system-based approach. With further development, such systems could play a crucial role in addressing global food insecurity and fostering long-term sustainability [6][8].

REFERENCES

- **[1]** United Nations. (2023). Global issues: Food. Retrieved from https://www.un.org/en/global-issues/food
- **[2]** International Telecommunication Union (ITU). (2023). Universal and meaningful connectivity by 2030. Retrieved from https://www.itu.int/en/mediacentre/Pages/PR-2023-09-12-universal-and-meaningful-connectivity-by-2030.aspx
- **[3]** Jha, K., Doshi, A., Patel, P., & Shah, M. (2022). A comprehensive review on automation in agriculture using artificial intelligence. *Computational Agriculture, 23*, 1-14.
- **[4]** Ghazal, S., Munir, A., & Qureshi, W. S. (2024). Computer vision in smart agriculture and precision farming: Techniques and applications. *Artificial Intelligence in Agriculture, 13*, 64-83.
- **[5]** Anjali, et al. (2024). State-of-the-art non-destructive approaches for maturity index determination in fruits and vegetables: Principles, applications, and future directions. *Food Production, Processing and Nutrition, 6*(56).
- **[6]** Subeesh, A., & Mehta, C. R. (2021). Automation and digitization of agriculture using artificial intelligence and internet of things. Artificial Intelligence in Agriculture, 5, 278-291.
- **[7]** Rane, J., Kumar, A., & Patel, P. (2024). Smart farming using artificial intelligence, machine learning, deep learning, and ChatGPT: Applications, opportunities, challenges, and future directions. In *Deep Science Research: Advanced Applications in AI* (pp. xx-xx). Deep Science Research Publications.
- **[8]** Adam, K. L. (2006). Community Supported Agriculture. ATTRA - National Sustainable Agriculture Information Service.