Design and Development of Solar-Powered Smart Dryer Agriculture and Food Products

[1] B.JAGADEESH, [2] A.PRABHAKAR,[3] N.DILEEP, [4] CH.TARUN KUMAR [5] CH.ESWAR GANESH

Mrs. JYOTHI UMMADI

Department of Mechanical Engineering
Welfare Institute of Technology & Management, Pinagadi , Visakhapatnam

Abstract—

The increasing demand for efficient and sustainable drying methods has led to the development of solar dryers, which utilize renewable energy sources for moisture removal in agricultural and food products. This project presents a solar-powered dryer that integrates a solar panel, 12V battery, 12V DC fan, L298 motor driver, Node MCU, and a multi-meter to optimize the drying process. The solar panel harnesses sunlight to generate electricity, which is stored in the 12V battery, ensuring continuous operation even in low-light conditions. The DC fan, controlled by the L298 motor driver and Node MCU, regulates airflow inside the dryer to enhance drying efficiency. The Node MCU also enables real-time monitoring and automation of the system, allowing users to adjust parameters remotely. Additionally, a multi-meter is used for performance analysis and voltage-current measurements. This system offers an eco-friendly, cost-effective, and efficient alternative to traditional drying methods, making it highly suitable for agricultural and small-scale industrial applications.

Key words: solar dryers***, Battery,Node mcu***

I. INTRODUCTION

Drying is a crucial process in preserving food, agricultural products, and other perishable materials by reducing moisture content, thereby preventing spoilage and extending shelf life. Traditional drying methods, such as open sun drying, are inefficient and prone to contamination, making solar dryers a more reliable and hygienic alternative.

This project focuses on designing and developing a solar-powered dryer that integrates a solar panel, 12V battery, 12V DC fan, L298 motor driver, Node MCU, and a multi-meter to create an energy-efficient drying system. The solar panel harnesses sunlight to generate electricity, which is stored in the 12V battery for uninterrupted operation. The DC fan, controlled by the L298 motor driver and Node MCU, facilitates proper airflow within the dryer, ensuring uniform drying. The Node MCU also enables remote monitoring and automation, optimizing the drying process for various applications. A multi-meter is included for real-time performance measurement, ensuring efficient energy usage.

By utilizing renewable solar energy, this system offers an eco-friendly, cost-effective, and sustainable solution for drying agricultural produce, herbs, spices, and other materials, reducing dependency on conventional energy sources while enhancing drying efficiency.

AIM OF THE PROJECT:

The aim of this project is to design and develop an efficient, cost-effective, and eco-friendly solar dryer that utilizes solar energy to dry agricultural and food products. By harnessing solar power, the system reduces dependence on conventional energy sources while ensuring faster, hygienic, and uniform drying. The project aims to improve the drying process, minimize post-harvest losses, and provide a sustainable drying solution for farmers, small-scale industries, and households.

OBJECTIVE OF THE PROJECT:

1. To harness solar energy for an efficient and eco-friendly drying process, reducing reliance on conventional electricity.
2. To design and develop a solar dryer that ensures uniform and faster drying of agricultural and food products.
3. To integrate a 12V battery for energy storage to enable continuous operation even in low sunlight conditions.
4. To use a 12V DC fan for proper airflow regulation, enhancing drying efficiency.
5. To implement an L298 motor driver and NodeMCU for automatic control and remote monitoring of the drying process.
6. To ensure real-time performance measurement using a multimeter for voltage and current analysis.
7. To provide a cost-effective and sustainable drying solution for small-scale farmers, households, and industries.
8. To reduce post-harvest losses by improving the drying process for food preservation.

SCOPE OF THE PROJECT:

1. **Renewable Energy Utilization:** The solar dryer operates using solar energy, making it an environmentally friendly and sustainable alternative to conventional drying methods.
2. **Efficient Drying Process:** The system ensures faster, uniform, and hygienic drying of agricultural products, food items, and other perishable materials.
3. **Energy Storage for Continuous Operation:** The integration of a 12V battery allows the system to function even in low sunlight conditions, ensuring uninterrupted drying.
4. **Automated Control and Monitoring:** The Node MCU and L298 motor driver enable smart automation, real-time monitoring, and remote control of the drying process.
5. **Versatile Applications:** The dryer can be used for agriculture (grains, fruits, vegetables, herbs), fisheries (drying fish), and medicinal plants, as well as for small-scale industries.
6. **Cost-Effective and Scalable:** The system is affordable, easy to maintain, and scalable, making it suitable for small farmers, households, and commercial applications.
7. **Performance Optimization:** A multi-meter is included to measure voltage and current, allowing efficient power management and system optimization.
8. **Reduction of Post-Harvest Losses:** By improving the drying process, the project helps in reducing food spoilage and wastage, enhancing food security and economic benefits.

3.COMPONENTS USED

HARDWARE DETAILS:

1. L298 MOTOR DRIVER
2. ESP8266 NODE MCU
3. SOLAR PANEL
4. DIODE
5. 12V BATTERY
6. 12V DC FAN
7. HUMIDITY SENSOE
8. MULTIMETER
9. JUMPER WIRES

SOFTWARE DETAILS:

1. ARDUINO IDE
2. EMBEDDED ‘C’

ARDUINO IDE:

The Arduino IDE (Integrated Development Environment) is a cross-platform software application designed for writing, compiling, and uploading code to Arduino microcontroller boards. It serves as the primary tool for developers, hobbyists, and students to create interactive electronics projects. With its intuitive interface and straightforward workflow, the Arduino IDE has become popular worldwide for prototyping and developing both simple and complex systems. The Arduino IDE stands out for its simplicity, flexibility, and extensive support network. It is ideal for beginners learning programming and electronics, as well as professionals developing advanced systems, making it a versatile tool in the world of embedded systems and IoT.

KEY COMPONENTS:

1. **Code Editor:**
   - The IDE provides a text editor where users write programs, known as "sketches," using a simplified version of the C++ programming language.
   - It features syntax highlighting, auto-indentation, and bracket matching, enhancing readability and reducing coding errors.
2. **Compiler:**
   - The built-in compiler translates human-readable code into machine language, ensuring compatibility with the microcontroller on the Arduino board.
3. **Uploader:**
   - Using a USB connection, the uploader transfers the compiled code from the computer to the Arduino board, enabling the microcontroller to execute the programmed instructions.
4. **Serial Monitor and Plotter:**
   - The Serial Monitor allows users to communicate with the Arduino board in real time, facilitating debugging and data exchange.
   - The Serial Plotter visualizes data from the board, making it easier to analyse sensor readings and other variables.
5. **Library Manager:**
   - A comprehensive library manager provides access to pre-written code libraries that simplify the use of sensors, displays, motors, and other hardware components.

FEATURES:

- **Cross-Platform Compatibility:** The Arduino IDE runs on Windows, macOS, and Linux.
- **Open Source:** As open-source software, it encourages collaboration, customization, and community contributions.
- **User-Friendly Interface:** Its straightforward design makes it accessible to beginners while still offering advanced features for experienced developers.
- **Extensive Community Support:** A vast online community shares code, tutorials, and troubleshooting tips, making it easier to learn and troubleshoot.

WORKING:

- **Writing Code:** Users write sketches using the code editor, often starting with built-in examples.
- **Compiling Code:** The IDE compiles the sketch, checking for syntax errors and ensuring compatibility with the selected Arduino board.
- **Uploading Code:** With a single click, the compiled code is uploaded to the Arduino board, enabling it to perform the desired tasks.
- **Testing and Debugging:** The Serial Monitor and Plotter help test the code, observe output data, and identify issues.

USAGE:

- Ideal for prototyping electronics projects, robotics, IoT devices, and automation systems.
- Widely used in education, hobbyist projects, and professional prototyping due to its ease of use and versatility.

Applications of Arduino IDE:

- **Education:** Widely used in schools and universities to teach programming and electronics.
- **Prototyping:** Essential for developing prototypes of IoT devices, robots, and home automation systems.
- **Hobbyist Projects:** Popular among makers for creating interactive art, DIY gadgets, and wearable technology.

EMBEDDED C

Embedded C is a specialized extension of the C programming language designed for programming embedded systems. Embedded systems are microcontroller-based devices that perform dedicated functions, such as medical devices, automotive systems, home appliances, and IoT devices. Unlike general-purpose computers, embedded systems operate with limited resources, making efficiency and reliability crucial. Embedded C helps developers write efficient and optimized code that directly interacts with hardware components. Embedded C plays a critical role in the development of embedded systems by combining the simplicity of C with hardware-specific functionalities. Its efficiency, low-level hardware control, and widespread industry adoption make it essential for designing devices that are reliable, responsive, and optimized for specific tasks.

CHARECTRISTICS:

- **Hardware-Oriented:** Embedded C allows direct manipulation of hardware components, such as input/output ports, timers, and memory.
- **Efficiency and Speed:** Programs are optimized to use minimal memory and processing power, essential for devices with limited resources.
- **Real-Time Operation:** Embedded systems often require real-time performance, where tasks must be executed within strict time limits.
- **Portability:** Embedded C code can be adapted to different microcontrollers with minimal changes, increasing its flexibility.
- **Low-Level Access:** The language provides low-level access to system memory and peripheral registers, allowing precise control over hardware.

STRUCTURE:

- **Preprocessor Directives:** Include header files specific to the microcontroller, such as <avr/io.h> for AVR microcontrollers.
- **Global Variables:** Store data that needs to be accessed throughout the program.
- **Main Function:** The main() function contains the core logic of the program and often runs in an infinite loop to maintain continuous operation.
- **Peripheral Initialization:** Configure hardware peripherals like timers, communication interfaces, and input/output ports.
- **Interrupt Service Routines (ISRs):** Handle hardware interrupts that respond to external events.

WORKING:

The working of Embedded C involves writing code that directly interacts with hardware components of embedded systems, such as microcontrollers, sensors, and actuators. Unlike standard C programming, Embedded C is designed to run on hardware with limited resources, ensuring efficiency, reliability, and real-time performance. The process typically follows these steps:

1. **Writing the Code:**
    - Developers write the program (called firmware) using Embedded C, which includes specific libraries and syntax for hardware interaction.
    - The code is usually structured with initialization routines, an infinite loop for continuous operation, and interrupt service routines (ISRs) to handle external events.

2. **Compiling the Code:**
    - The written code is compiled using a cross-compiler, which translates the human-readable C code into machine code (binary format) that the microcontroller understands.
    - The compiler is specific to the target microcontroller architecture (e.g., AVR-GCC for AVR microcontrollers or Keil C51 for 8051 microcontrollers).

3. **Linking and Optimization:**
    - The linker combines different code modules into a single executable file, optimizing the code to minimize memory usage and maximize performance.
    - Optimization is crucial since embedded systems often have limited memory and processing power.

4. **Generating the Hex File:**
    - The final output of the compilation process is a HEX file, which contains the machine code in hexadecimal format.
    - This file is ready to be uploaded to the microcontroller.

5. **Uploading the Code to the Microcontroller:**
    - The HEX file is transferred to the microcontroller's flash memory using a programmer device or a USB connection.
    - For example, in Arduino, this process is handled by the Arduino IDE using a built-in bootloader.

6. **Executing the Program:**
    - Once uploaded, the microcontroller reads the machine code from its memory and executes the instructions continuously.
    - The main function typically runs in an infinite loop to ensure the system responds continuously to inputs and performs the desired tasks.

7. **Real-Time Interaction with Hardware:**
    - The program directly interacts with hardware components using peripheral registers, input/output ports, and communication interfaces (e.g., SPI, I2C, UART).
    - Interrupts allow the system to respond immediately to external events, ensuring real-time performance.

Real-Time Execution and Interrupts:

- Embedded systems often need to respond to external events within strict time limits.
- Interrupts allow the microcontroller to pause the main program and execute specific code when an external event occurs (e.g., a button press or sensor signal).

ADVANTAGES:

- **Simplicity and Readability:** Easy to learn and read, especially for those familiar with standard C.
- **Efficiency:** Generates compact, fast-executing code suitable for resource-limited systems.
- **Portability:** Code can be reused across different microcontrollers with minimal modifications.
- **Direct Hardware Control:** Provides precise control over hardware components, enabling efficient system design.

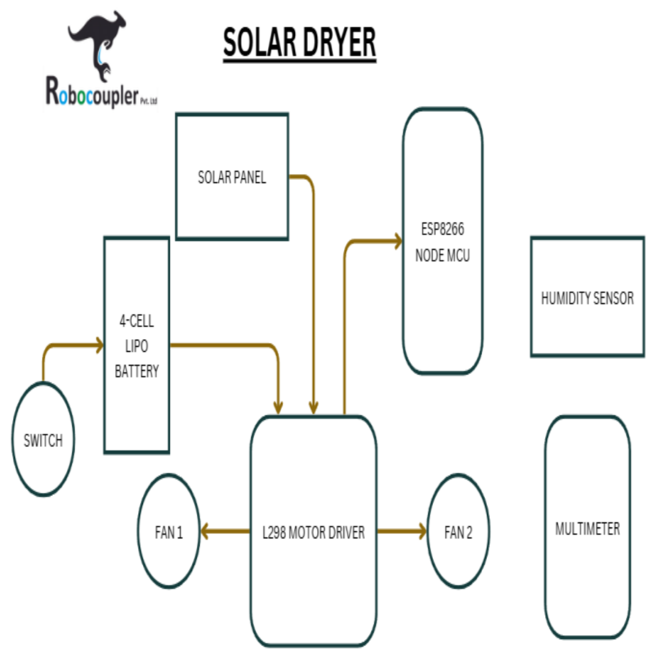
- **Wide Industry Adoption:** Used extensively in automotive, healthcare, consumer electronics, and IoT applications.

APPLICATIONS:

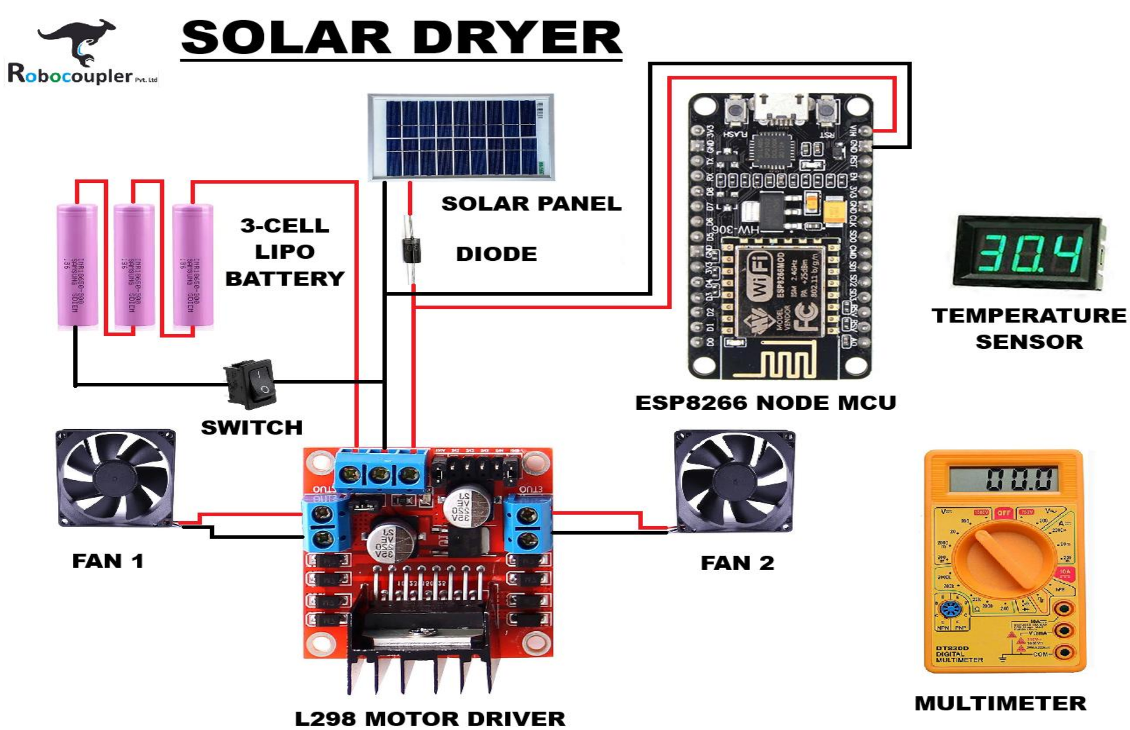
- **Automotive Systems:** Engine control units (ECUs), anti-lock braking systems (ABS), and airbags.
- **Medical Devices:** Pacemakers, blood pressure monitors, and ventilators.
- **Consumer Electronics:** Microwave ovens, washing machines, and smart home devices.
- **Industrial Automation:** Motor control, process monitoring, and robotics.
- **IoT Devices:** Sensors, actuators, and wireless communication modules.

4.ARCHITECTURE

BLOCK DIAGRAM:



CIRCUIT DIADGRAM:



WORKING:

Solar Panel:

- The solar panel is the primary power source, converting solar energy into electrical energy.
- The generated DC voltage is used to charge the battery and power the system.
- A diode is placed in series to prevent reverse current flow from the battery to the panel.

3-Cell LiPo Battery:

- A 3-cell LiPo (Lithium Polymer) battery is used to store excess solar energy, ensuring uninterrupted operation during low sunlight or nighttime.
- The battery is connected to the circuit through a switch, allowing manual control of power supply.

ESP8266 Node MCU (Wi-Fi Microcontroller):

- The ESP8266 Node MCU acts as the brain of the system, providing automation and remote monitoring.
- It receives temperature sensor data, controls fans, and communicates with external devices via Wi-Fi (IoT-based monitoring).
- It is powered by the solar panel and battery setup.

Temperature Sensor:

- The temperature sensor continuously monitors the drying chamber's temperature.
- The sensor data is sent to the Node MCU, which adjusts fan speed accordingly for optimal drying conditions.
- The temperature is displayed on a digital screen for real-time monitoring.

L298 Motor Driver:

- The L298 motor driver is used to control the 12V DC fans.
- It acts as an interface between the Node MCU and the fans, providing the necessary power and control.
- The Node MCU sends signals to the motor driver to adjust fan speed based on drying requirements.

12V DC Fans (Fan 1 & Fan 2):

- Two 12V DC fans are used to maintain proper air circulation inside the drying chamber, ensuring uniform drying.
- The fans remove moisture-laden air and replace it with dry air, reducing drying time.
- The speed of the fans is controlled by the Node MCU and L298 motor driver.

Multi-meter:

- A digital multi-meter is connected to measure voltage, current, and power consumption of the system.
- This helps in analysing the performance and efficiency of the solar dryer.

Power Generation & Storage:

- The solar panel generates electricity, which powers the system and charges the LiPo battery for backup operation.

Temperature Monitoring:

- The temperature sensor continuously measures the drying chamber temperature and sends the data to the Node MCU.

Fan Control for Air Circulation:

- The Node MCU processes temperature data and sends control signals to the L298 motor driver.
- The motor driver regulates the speed of the DC fans to ensure optimal airflow and faster drying.

Real-Time Monitoring & Energy Measurement:

- The temperature display shows the real-time temperature of the drying chamber.
- The multi-meter helps in monitoring power usage and system efficiency.

RESULTS

1. The solar panel successfully generated sufficient power under sunlight conditions, charging the 12V LiPo battery efficiently.
2. Backup power from the battery allowed the system to continue functioning during low sunlight hours.
3. The multi-meter readings confirmed stable voltage and current supply, ensuring efficient energy use.
4. The DC fans provided consistent airflow, reducing drying time compared to traditional open sun drying.
5. The L298 motor driver effectively controlled fan speed, adjusting airflow based on temperature conditions.
6. The drying process was faster and more uniform, preventing Mold growth and spoilage.
7. The temperature sensor accurately measured and displayed real-time drying chamber temperature.
8. The Node MCU processed temperature data and adjusted fan speed automatically, optimizing the drying conditions.
9. Remote monitoring via Wi-Fi (IoT) allowed real-time data tracking, enhancing user convenience.
10. The system operated efficiently, consuming minimal power due to the optimized use of solar energy.

ADVANTAGES AND RESULTS

ADVANTAGES:

1. **Energy-Efficient:** Utilizes solar energy, reducing dependency on conventional power sources.
2. **Renewable:** Low operational costs due to free sunlight availability.
3. **Uniform Drying:** Eliminates moisture buildup, preventing fungal and bacterial growth.
4. **Automated:** Node MCU automates fan speed based on temperature variations, reducing manual intervention.
5. **Contamination-Free Drying:** Unlike traditional open drying, this system protects food from dust, insects, and environmental pollutants.
6. **Cost-Effective:** Long lifespan of components like solar panels, batteries, and fans.

FUTURE SCOPE:

1. Addition of humidity sensors for more precise drying control.
2. Development of a mobile app for real-time monitoring, notifications, and control.
3. Combining solar energy with wind or biogas for uninterrupted power.
4. Scaling the system for commercial food processing and agricultural

CONCLUSION

The solar dryer system utilizing a solar panel, 12V battery, 12V DC fan, L298 motor driver, Node MCU, and multi-meter provides an efficient, eco-friendly, and automated solution for drying agricultural and food products. By harnessing solar energy, the system significantly reduces dependency on conventional power sources, making it cost-effective and sustainable.

The integration of Node MCU for automation ensures smart control of airflow and temperature, optimizing the drying process and reducing human intervention. The 12V battery backup allows uninterrupted operation, even in low sunlight conditions, improving reliability. Multi-meter analysis confirms the efficient energy utilization, making the system a viable alternative to traditional drying methods.

Compared to conventional sun drying, this solar dryer enhances drying efficiency, reduces drying time, minimizes contamination risks, and ensures uniform drying. With IoT integration, the system can be further developed for remote monitoring and control, making it ideal for households, small-scale farmers, and large-scale agricultural applications.

The solar dryer system is a step toward sustainable energy solutions, promoting clean technology in food preservation and agricultural processing. Future improvements, such as humidity sensors, AI-based automation, and mobile app integration, can further enhance its efficiency and usability.

This project demonstrates the potential of renewable energy in modern applications, proving that solar-powered innovations can lead to energy-efficient, cost-saving, and eco-friendly solutions for a better future.

REFERENCES

1. Heat and Mass transfer by R .K.Rajput.
2. Heat and Mass transfer by Books Indiapublication.
3. Alternate Energy Sources by Tech-Max publication.
4. Thermodynamics by S. chand publication.
5. www.iris.lib.neu.edu
6. www.independentscience.blogspot.com
7. S. B. Heaters, Nirmalnagar, Bhavnagar.
8. Raj furniture, pirchalla sheri, Bhavnagar.