Journal on Electronic and Automation Engineering

Vol: 4(2), June 2025

REST Publisher; ISSN: 2583-6951 (Online)

Website: https://restpublisher.com/journals/jeae/

DOI: https://doi.org/10.46632/jeae/4/2/42

Solar-Powered Laser Security System for Agriculture

***B. Hoglah Leena, K. Aravindh, M. Vineetha, M. Navya Sri**

Nalla Malla Reddy Engineering College, Hyderabad, Telangana, India.

*Corresponding Author Email: leenah.ece@nmrec.edu.in

Abstract: *The Solar-Powered Laser Security System for Agriculture is a smart solution that uses solar energy and laser technology to protect crops from animal intrusions. It employs solar panels to power laser beams and photodiodes, which detect interruptions caused by animals. When a beam is broken, the ESP32 microcontroller processes the data and triggers alerts via Wi-Fi or GSM, notifying farmers through SMS or app notifications. The system is energy-efficient, supports remote monitoring, and remains operational even in off-grid areas, making it highly effective for agricultural security.*

1. INTRODUCTION

Agriculture is increasingly facing challenges related to security, with theft, vandalism, and wildlife intrusions threatening crops and livestock. To address these issues, a solar powered laser security system offers an innovative solution. This system uses a laser perimeter and advanced sensor technologies to detect intrusions in real time, sending immediate alerts to farmers. Powered by renewable solar energy and controlled by the efficient ESP32 microcontroller, the system is cost effective, energy efficient, and scalable, making it an ideal solution for modern agricultural security needs. It not only ensures the safety of crops but also promotes sustainability by reducing dependence on conventional power sources.

The objectives of the project include:

1. Design a solar-powered laser security system for agriculture to detect intrusions.
2. Integrate the ESP32 microcontroller for real-time monitoring and control.
3. Create a solar-powered perimeter using laser beams to detect unauthorized access.
4. Implement Wi-Fi/Bluetooth connectivity for remote notifications to farmers.
5. Develop a cost-effective and scalable solution for agricultural security.

Literature Review: The integration of solar power into security systems, especially in remote agricultural areas, has gained significant attention due to its sustainability and energy independence. Several studies support this innovation: Solar Energy: Principles and Applications (2000) details solar energy fundamentals and real-world applications; Practical Electronics for Inventors (2016) explains core electronics and system design; and The Internet of Things: Approach (2014) explores IoT platforms like Arduino and ESP32 for remote monitoring. Additionally, Microcontroller-based Projects for Beginners (2011) provides hands-on experience in embedded systems, while Smart Agriculture (2019) shows how technologies like IoT and machine learning optimize farming practices. Lastly, Renewable Energy: Power for a Sustainable Future (2012) offers a broad overview of renewable systems and their environmental benefits. Together, these resources underpin the development of solar-powered laser security systems using ESP32 for agricultural protection.

2. EMBEDDED SYSTEMS

Embedded Systems: An embedded system is a specialized computer system designed to perform dedicated tasks, often under real-time constraints. Unlike general-purpose computers, embedded systems are part of larger devices and are optimized for specific functions using microcontrollers or digital signal processors (DSPs). These systems control many modern devices, from digital watches to industrial controllers and even components of air traffic control systems. Their focused purpose allows for reduced size, cost, and increased reliability. While some are simple and portable, others are complex, comprising multiple processors and extensive hardware. Although the term "embedded system" can be broad, its essence lies in performing specific functions efficiently, with limited flexibility compared to general-purpose systems.

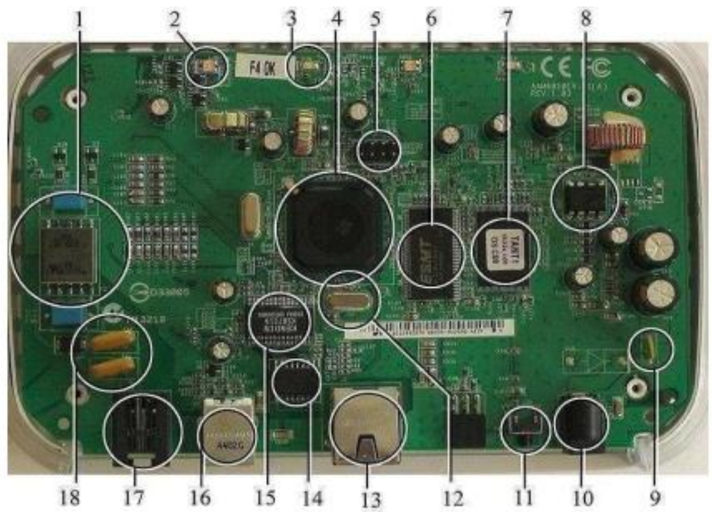

FIGURE 1. A modern example of an embedded system

Embedded systems programming differs significantly from traditional PC programming. It often resembles the resource-constrained programming of early computing ERAS; as embedded hardware is typically selected for cost-efficiency rather than convenience. Adding just a dollar per unit in hardware can lead to millions in additional costs, making it more economical to invest in programmer time instead. Consequently, developers must work with limited processing power and memory, while still ensuring highly efficient and reliable performance. Key components such as the microprocessor, RAM, and flash memory are tightly integrated, and the field presents unique challenges that demand specialized skills and considerations.

History: In the 1930s and 1940s, early computers were often dedicated to single tasks but were too large and costly for widespread embedded use. As technology evolved, programmable controllers emerged from electromechanical sequencers and solid-state devices. A major milestone was the Apollo Guidance Computer, developed at MIT, which used integrated circuits to reduce size and weight—considered highly risky at the time. Another early example was the Autonetics D17, used in the Minuteman missile system in 1961. It was later replaced in 1966 by a version featuring the first high-volume use of integrated circuits.

Tools: Embedded development makes up a small fraction of total programming. There're also many embedded architectures, unlike the PC world where 1 instruction set rules, and the Unix world where there's only 3 or 4 major ones. This means that the tools are more expensive. It also means that they're lowering featured, and less developed. On a major embedded project, at some point you will almost always find a compiler bug of some sort. Debugging tools are another issue. Since you can't always run general programs on your embedded processor, you can't always run a debugger on it. This makes fixing your program difficult. Special hardware such as JTAG ports can overcome this issue in part. However, if you stop on a breakpoint when your system is controlling real world hardware (such as a motor), permanent equipment damage can occur. As a result, people doing embedded programming quickly become masters at using serial IO channels and error message style debugging.

Resources: To save costs, embedded systems frequently have the cheapest processors that can do the job. This means your programs need to be written as efficiently as possible. When dealing with large data sets, issues like memory cache misses that never matter in PC programming can hurt you. Luckily, this won't happen too often- use reasonably efficient algorithms to start and optimize only when necessary. Of course, normal profilers won't work well, due to the same reason debuggers don't work well. Memory is also an issue. For the same cost savings reasons, embedded systems usually have the least memory they can get away with. That means their algorithms must be memory efficient (unlike in PC programs, you will frequently sacrifice processor time for memory, rather than the reverse). It also means you can't afford to leak memory. Embedded applications generally use deterministic memory techniques and avoid the default "new" and "malloc" functions, so that leaks can be found and eliminated more easily. Other resources programmers expect may not even exist. For example, most embedded processors do not have hardware FPUs (Floating-Point Processing Unit). These resources either need to be emulated in software or avoided altogether.

Real Time Issues: Embedded systems frequently control hardware and must be able to respond to them in real time. Failure to do so could cause inaccuracy in measurements, or even damage hardware such as motors. This is made even more difficult by the lack of resources available. Almost all embedded systems need to be able to prioritize some tasks over others, and to be able to put off/skip low priority tasks such as UI in favor of high priority tasks like hardware control.

Need for Embedded Systems: The uses of embedded systems are virtually limitless, because every day new products are introduced into the market that utilizes embedded computers in novel ways. In recent years, hardware such as microprocessors, microcontrollers, and FPGA chips have become much cheaper. So, when implementing a new form of control, it's wiser to just buy the generic chip and write your own custom software for it. Producing a custom-made chip to handle a particular task or set of tasks costs far more time and money. Many embedded computers even come with extensive libraries, so that "writing your own software" becomes a very trivial task indeed. From an implementation viewpoint, there is a major difference between a computer and an embedded system. Embedded systems are often required to provide Real-Time response. The main elements that make embedded systems unique are its reliability and ease in debugging.

Debugging: Embedded debugging can be done at various levels using tools like interactive shells (e.g., Forth), external logs via serial ports, or advanced tools like in-circuit debuggers (ICD) and emulators. ICDs connect through JTAG interfaces to control processor functions, while emulators can simulate full hardware for PC-based debugging. Debugging strategies differ based on whether the system is software-centric or peripheral-driven (e.g., using DSPs or FPGAs). Multi-core systems add complexity, requiring careful synchronization and sometimes low-level signal analysis using logic analyzers to monitor inter-core data traffic.

Reliability: Embedded systems often reside in machines that are expected to run continuously for years without errors and in some cases recover by them if an error occurs. Therefore, the software Embedded systems are designed for high reliability, especially in environments where shutdowns are unsafe, costly, or inaccessible—such as in aircraft, space systems, or industrial controls. To ensure continuous operation, they avoid moving mechanical parts and are tested rigorously. Common reliability strategies include watchdog timers to reset the system in case of software failure, redundant subsystems, and software limp modes that offer partial functionality. Secure architectures like Trusted Computing Base (TCB) and Embedded Hypervisors also enhance system reliability by isolating faults and preventing them from affecting other components.

Explanation of Embedded Systems: Embedded systems use various software architectures, such as simple control loops, interrupt-controlled systems, cooperative multitasking, and primitive multitasking. Simple control loops run continuous cycles of subroutine calls to manage hardware and software. Interrupt-controlled systems are event-driven, triggered by interrupts such as timers or external signals, providing low-latency event handling. Cooperative multitasking systems execute tasks that yield control voluntarily, making software additions easier. Primitive multitasking uses timers to switch tasks and is the basis for operating systems, requiring careful synchronization to avoid conflicts in data access. For larger systems, real-time operating systems (RTOS) are often used, though smaller systems may not afford the overhead. Microkernels and exokernels offer flexibility for task management and inter-task communication based on performance needs.

Embedded System Types:

1. Standalone Embedded Systems: These systems process inputs like electrical signals or human commands (e.g., pressing a button) and generate outputs without external dependencies.
2. Real-time Embedded Systems: These systems must perform tasks within strict time constraints. Hard real-time systems have absolute deadlines, where delays could cause damage (e.g., opening a valve within 30 ms). Soft real-time systems allow for delays without causing harm (e.g., remote control response delays).

3. Network Communication Embedded Systems: These systems enable communication over networks, such as using a web camera connected to the internet to send images or videos. For example, a door lock system can capture an image when a person is near, alerting a user on their computer to open the lock remotely.

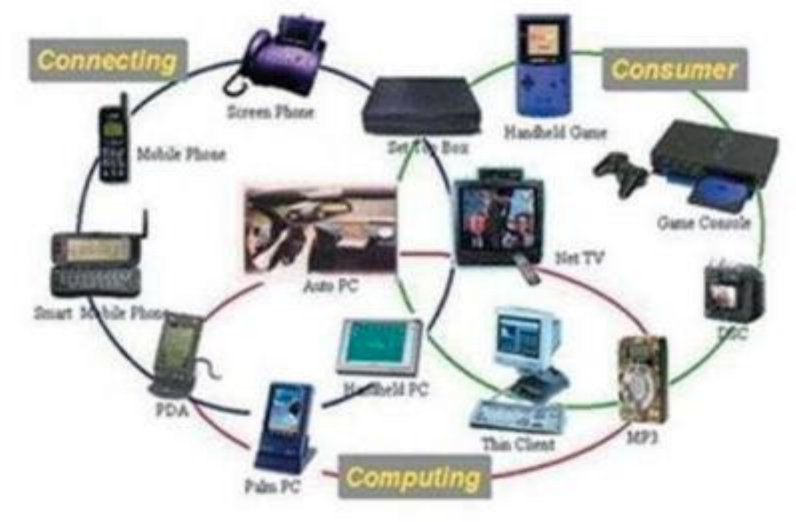

FIGURE 2. Network communication embedded systems

Different types of processing units:

The central processing unit (CPU) can be one of the following: microprocessor, microcontroller, or digital signal processor (DSP). Among these, microcontrollers are low-cost processors, and their key advantage is the integration of various components like memory, serial communication interfaces, and analog-to-digital converters on a single chip. This reduces the need for external components, making them ideal for specific applications with minimal external connections.

4. HARDWARE DESCRIPTION

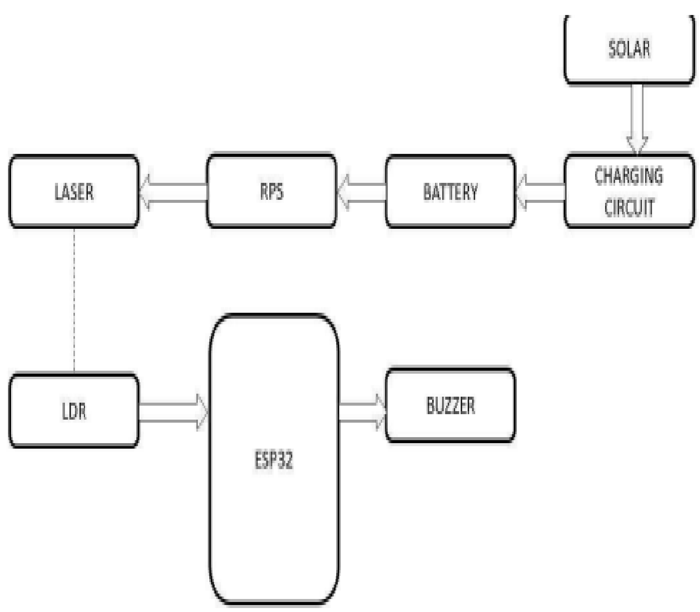

FIGURE 3. Block diagram of Solar-Powered Laser Security System for Agriculture using ESP32

FIGURE 4. ESP32 CAMERA

FIGURE 5. Battery

BUZZER:

A piezoelectric sound component typically consists of a piezoelectric diaphragm, which includes a piezoelectric ceramic plate with electrodes on both sides and a metal plate (usually brass or stainless steel). When a DC voltage is applied between the electrodes, mechanical distortion occurs due to the piezoelectric effect, causing the diaphragm to bend. This bending creates sound waves when AC voltage is applied. To interface a buzzer, a standard transistor circuit is used, ensuring the 0V rails of each power supply are connected for a common reference. Piezo sounders consume less current than buzzers and can produce multiple tones, whereas buzzers generate a single tone.

Handling and Design Notes:

1. Ensure proper feedback voltage control to avoid continuous sound.
2. Follow the recommended circuit design for stable oscillation.
3. Direct power switching should be used for proper operation.
4. The self-drive circuit is already integrated into the piezo buzzer, eliminating the need for an additional driving circuit.
5. Maintain the rated voltage (3.0 to 20Vdc) for optimal performance.
6. Avoid placing resistors in series with the power source to prevent abnormal oscillations. Use a capacitor for adjusting sound pressure.
7. Do not obstruct the sound emitting hole.
8. Ensure no obstacles are within 15mm of the sound release hole for proper sound emission.

FIGURE 6. Picture of Buzzer

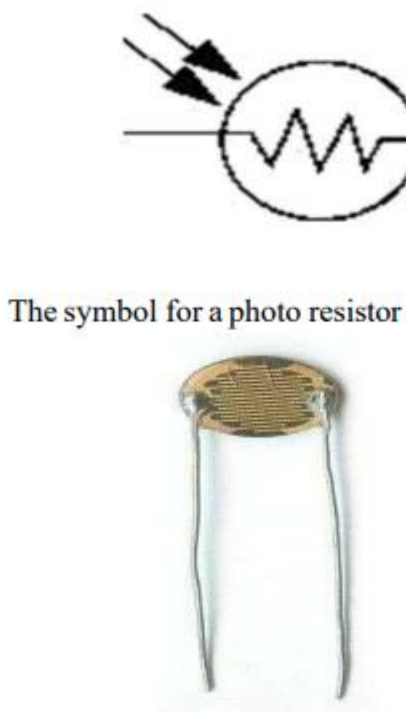

FIGURE 7. LDR

Light Dependent Resistor – LDR:

Two cadmium sulphide (cds) photoconductive cells with spectral responses like that of the human eye. The cell resistance falls with increasing light intensity. Applications include smoke detection, automatic lighting control, batch counting and burglar alarm systems.

FIGURE 8. Photo Diode

5. SOFTWARE DESCRIPTION

To boot load the ATmega328 on a breadboard using an Arduino UNO, you will need a few components: an ATmega328P or ATmega328PU microcontroller, an Arduino UNO, a breadboard, jumper wires, and the Arduino IDE installed on your PC. Start by wiring the Arduino UNO to the ATmega328 on the breadboard, connecting specific pins for Reset, MOSI, MISO, SCK, GND, and VCC. Once the connections are made, open the Arduino IDE and select the Arduino UNO board and "Arduino as ISP" programmer under the tools menu. To load the bootloader onto the ATmega328PU (if it's not already pre-installed), go to Tools Burn Bootloader. After successfully burning the bootloader, you can upload sketches to the ATmega328 using the Upload Using Programmer option in the IDE. This setup enables you to program the ATmega328 just like an Arduino board, using the Arduino UNO as a programmer.

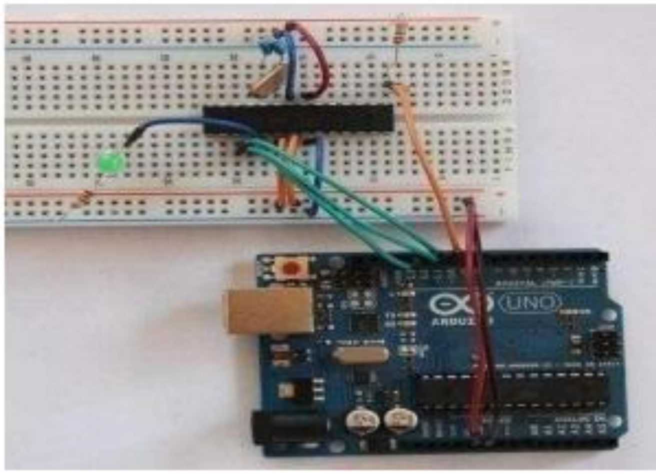

FIGURE 9. Arduino UNO Step

FIGURE 10. Program your Arduino UNO

To program the Arduino UNO to act as an ISP (In-System Programmer) for burning the bootloader onto the breadboard chip, first connect the Arduino UNO to your PC via USB, but ensure that it is not connected to the breadboard yet. Open the Arduino IDE on your PC, then navigate to **File > Examples > 11.ArduinoISP > ArduinoISP** to open the ArduinoISP sketch. If you're using Arduino IDE version 1.0, make a modification in the sketch by searching for the line delay(40); and changing it to delay(20);, as this adjusts the timing for the heartbeat function. Next, select **Arduino UNO** under the **Tools > Board** menu and choose **Arduino as ISP** from the **Tools > Programmer** menu. Finally, upload the modified ArduinoISP sketch to the Arduino UNO. Once the upload is complete, the Arduino UNO will be ready to act as an ISP for burning the bootloader onto the breadboard chip, allowing you to connect it to the breadboard and proceed with further programming.

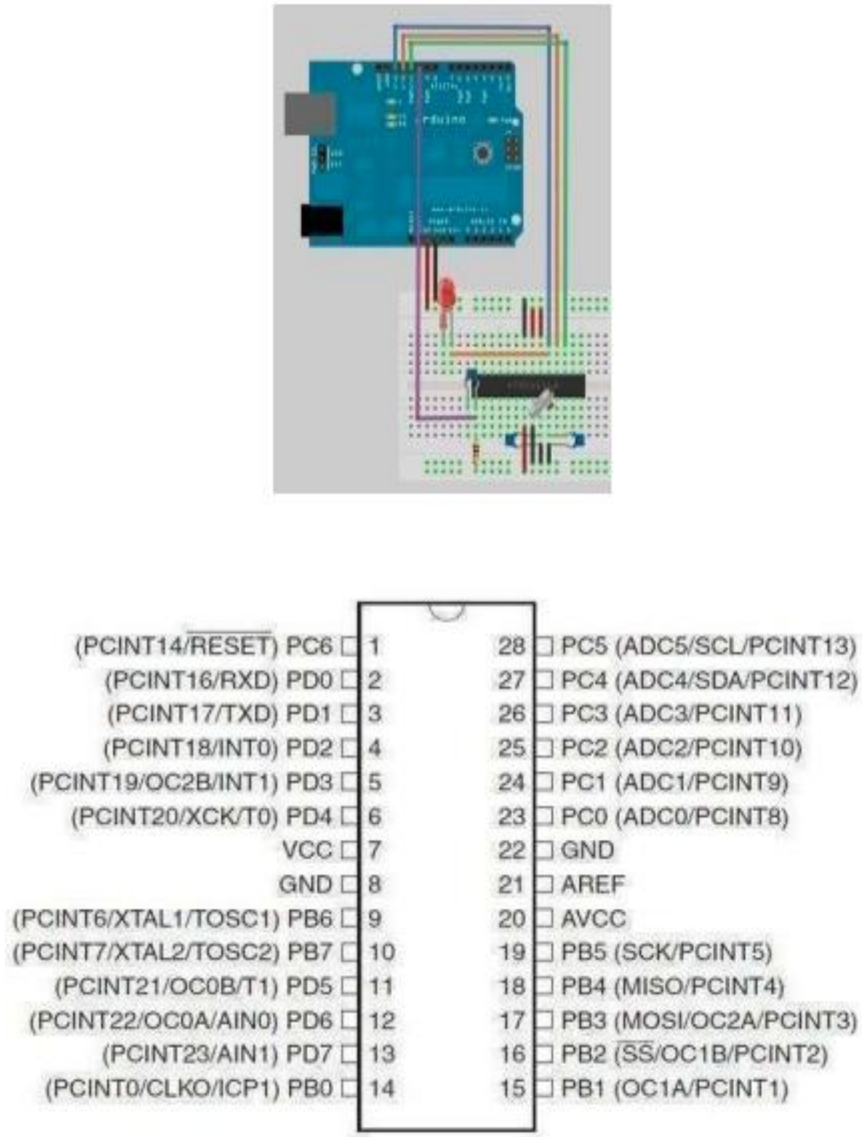

FIGURE 11. Connect your ATmega328

6. ADVANTAGES AND DISADVANTAGES

Advantages:

- Energy-efficient solar power reduces electricity costs.
- Cost-effective and affordable for farmers.
- Real-time intrusion detection with instant alerts.
- Remote monitoring via Wi-Fi/Bluetooth connectivity.
- Scalable solution for farms of varying sizes.
- Reduced crop damage and loss from unauthorized access.
-

Disadvantages:

- Limited by sunlight availability, with performance affected in cloudy weather.
- Initial setup cost for solar panels and components.
- Potential range limitations for laser detection in large areas.

Applications:

- Home Security: Protecting homes and residential properties from intruders.
- Office Security: Securing offices and commercial buildings to prevent theft.
- Retail Security: Monitoring retail stores to prevent shoplifting and ensure safety.
- Remote Monitoring: Enabling remote monitoring of properties via smart mobile devices.
- Outdoor Security: Monitoring outdoor areas like gardens, driveways, and parking lots to deter intruders and ensure safety.

7. RESULTS

The solar-powered laser security system provides an innovative, cost-effective solution for agricultural security. By using solar energy, it offers an environmentally friendly and energy efficient way to detect intrusions in real-time. The system's scalability allows it to be tailored to various farm sizes, while its wireless connectivity ensures easy monitoring. With reduced maintenance requirements and the ability to prevent crop loss, this technology enhances both security and sustainability in modern agriculture. Overall, it provides a reliable, low-cost, and advanced solution for protecting agricultural assets.

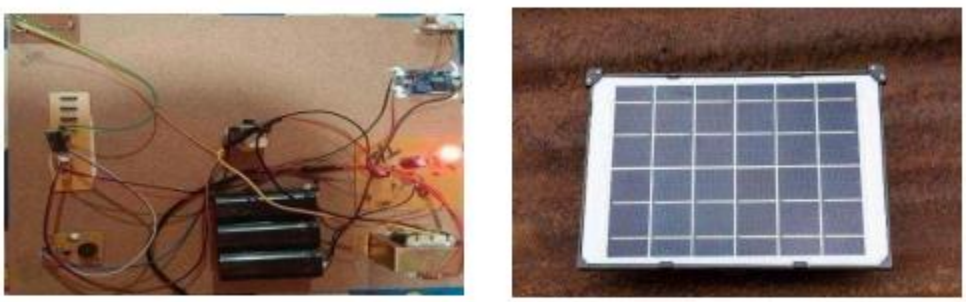

FIGURE 9. Demonstration of the project, Solar

FIGURE 10. Intruder alert, ESP32'S captured image

8. CONCLUSION

Integrating features of all the hardware components used have been developed in it. Presence of every module has been reasoned out and placed carefully, thus contributing to the best working of the unit. Secondly, using highly advanced IC's with the help of growing technology, the project has been successfully implemented. Thus the project has been successfully designed and tested.

Future Scope: The future scope of this solar-powered laser security system in agriculture is promising, with potential for further integration of advanced technologies like AI and machine learning to enhance intrusion detection and predictive analytics. The system could be expanded to include features such as facial recognition for animal control or automated responses to threats. Additionally, its scalability and adaptability could make it suitable for large-scale farms, commercial agriculture, and even urban farming. As renewable energy technology continues to evolve, the system's efficiency and affordability will likely improve, making it an increasingly vital tool for sustainable agricultural security worldwide.

REFERENCES

[1]. Electronics For You. www.electronicsforu.com. International Research Journal of Engineering and Technology (IRJET). Volume 4, Issue 3, March 2017. eISSN: 2395-0056. Available at: www.irjet.net. International Journal of Computer Applications. March 2015.

[2]. Amit Gupta, M. Senthil Kumar, M. Raman Kumar and D. Hemanth Kumar, *"Deep Learning Technique Used for Tomato and Potato Plant Leaf Disease Classification and Detection",* 2023 International Conference on Smart Systems for applications in Electrical Sciences (ICSSES), July 2023.

[3]. Amit Gupta, Abha Dargar, Abdul Majid, Shashi Kant Dargar, M. Senthil Kumar and M. Raju, *"Markov Chain Model Used in Agricultural Yield Predictions Utilizing on Indian Agriculture",* 2023 IEEE World Conference on Applied Intelligence and Computing (AIC), July 2023.

[4]. M. Senthil Kumar and Ashish Chaturvedi, *"A Novel Enhanced Coverage Optimization Algorithm for Effectively Solving Energy Optimization Problem in WSN",* Research Journal of Applied Sciences, Engineering and Technology (RJASET). (Issue 4, Vol.7, January 2014, ISSN: 2040 – 7459 & e-ISSN: 2040 – 7467).

[5]. International Journal of Emerging Technology and Advanced Engineering (IJETAE). Volume 5, Issue 10, October 2015. ISSN: 2250-2459. Available at: www.ijetae.com.

[6]. M. Senthil Kumar and Ashish Chaturvedi, *"Energy-Efficient Coverage and Prolongs for Network Lifetime of WSN using MCP",* European Journal of Scientific Research (EJSR). (Vol.95, No.2, January 2013, ISSN: 1450 – 216X / 1450 – 202X).

[7]. Senthilkumar Meyyappan, G. Lava Kumar, G. Niharika and G. Chakradhar, *"Cellular Network Signal Strength Analyser",* Journal on Electronic and Automation Engineering, Vol. 4(1), March 2025, pp. 165-174.

[8]. M. Senthil Kumar and C. Sridhathan, *"Impact of Mobility on the Routine of Enhanced – DSDV Protocol in Mobile Ad-hoc Networks",* International Journal of Applied Engineering Research (IJAER). (Vol.13, No.14, 2018, PP 11674-11679, ISSN: 0973-4562).

[9]. Yauri, R., Campos, E., Yalico, R., & Gamero, V. (2023). Development of an Electronic Bird Repellent System using Sound Emission. Published on ResearchGate, May 19, 2023.

[10]. C.I. Vimalarani and M. Senthil Kumar, *"Energy Efficient PCP Protocol for k-Coverage in Sensor Networks",* IEEE International Conference on Computational Intelligence and Computing Research, IEEE Proceedings, 2010.

[11]. M. Senthil Kumar, *"Energy Efficient Techniques for Transmission of Data in Wireless Sensor Networks",* Journal of Computing Technologies (JCT). (Vol.5, Issue 2, February 2016, ISSN: 2278 – 3814).

[12]. C. Sridhathan and M. Senthil Kumar, *"Plant Infection Detection using Image Processing",* International Journal of Modern Engineering Research (IJMER). (Vol.8, Issue 7, 2018, PP 13-16, ISSN: 2249 – 6645).

[13]. Lee, C. W., Muminov, A., Ko, M.-C., Oh, H.J., Lee, J. D., Kwon, Y. A., Na, D., Heo, S.-P., & Jeon, H. S. (2021). Anti-Adaptive Harmful Birds Repelling Method Based on Reinforcement Learning Approach. IEEE Journals & Magazine, IEEE Xplore.

[14]. M. Senthil Kumar and L. Praveen, *"An Assuring Approach for Tree-Based Routing Topology in WSNs"*, International Journal of Emerging Trends in Engineering and Development (IJETED). (Issue 3, Vol.6, November 2013, ISSN: 2249 – 6149).

[15]. M. Senthil Kumar and M. Gopinath, *"An Efficient Polynomial Pool-Based Scheme for Distributed Heterogeneous WSNs"*, International Journal of Modern Engineering Research (IJMER). (Vol.3, Issue 6, Nov-Dec.2013, PP 3328-3335, ISSN: 2249-6645).

[16]. M. Senthil Kumar, R. Karthik and I. Rabeek Raja, *"An Efficient Approach for Increasing Power Optimization in Mobile Ad-Hoc Networks"*, International Journal of Engineering Research and Technology (IJERT). (Vol.3, Issue 2, February 2014).

[17]. Ferreira Lima, M. C., Damascena de Almeida Leandro, M. E., Valero, C., Pereira Coronel, L. C., & Gonçalves Bazzo, C. O. (2020). Automatic Detection and Monitoring of Insect Pests, Published on May 9, 2020.

[18]. Senthilkumar Meyyappan and N. Selvamuthukumaran, *"Network Selection in Heterogeneous Wireless Systems using GRA Method",* Journal on Electronic and Automation Engineering, Vol. 4(1), March 2025, pp. 127-132.

[19]. Senthilkumar Meyyappan, A. Bharath Naik, A. Uma Sai and Ch. Keerthi, *"Improving Weather Forecasting Accuracy Using Machine Learning",* Journal on Electronic and Automation Engineering, Vol. 2(4), December 2023, pp. 9-18.

[20]. Senthilkumar Meyyappan, Kalyan Kasturi, G. Vijaya Lakshmi, J. Srinija Reddy and K. Grace Sampoorna, *"Improvement of LEACH Protocol for Enhancing Features of WSN",* Journal on Electronic and Automation Engineering, Vol. 2(4), December 2023, pp. 19-26.

[21]. Arowolo, O., Adekunle, A., & AdeOmowaye, J. (2020). A Real-Time Image Processing Bird Repellent System Using Raspberry Pi. FUOYE Journal of Engineering and Technology (FUOYEJET), Vol. 5, Issue 2, September 2020. ISSN: 2579-0625 (Online), 2579-0617 (Print).
[22]. M. Kavitha, T. Maheshwaran and M. Senthil Kumar, *"Secure Routing in MANETs with Key Management",* International Journal on Engineering Technology and Sciences (IJETS). (Vol.1, Issue 6, October 2014, ISSN (P): 2349 – 3968, ISSN (O): 2349 - 3976).
[23]. Senthilkumar Meyyappan, K. Susmitha, K. Vaishnavi and M. Sai Rao, *"Condition Based Monitoring and Maintenance System for Underground Metro Stations",* Journal on Electronic and Automation Engineering, Vol. 4(1), March 2025, pp. 175-182.
[24]. Nagaraju, P., Madhavi, A., Bhavani, B., Sunilkumar, G., & Vamsi, A. (2019). Solar- Based Birds Repeller to Protect Crops from Birds and Animals. International Journal of Research in Engineering, IT and Social Science, May 2019. ISSN: 2250-058
[25]. Yed,R.,& Javed,A. (2024). Smart Animal Repellent System Agriculture.International Journal of Science Technology and Management (IJSTM). Electronics ForYou. www.electronicsforu.com.

