International Scientific Journal of Engineering and Management ISSN: 2583-6129
Volume: 03 Issue: 03 | March – 2024 DOI: 10.55041/ISJEM01356
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Arduino Controlled Battery Charger

Prof Tejaswini Thakur, Prof .Sumit Thakre Harshada Jagate , Divya Bandgar ,Mayuresh Dhadawale

[1]*Prof.Tejaswini Thakur,Electrical Department & Zeal College of Engineering,Pune*
[2]*Prof.Sumit Thakre,Electrical Department & Zeal College of Engineering,Pune*
[3] *Harshada Jagate ,Electrical Department & Zeal College of Engineering,Pune*
[4] *Divya Bandgar ,Electrical Department & Zeal College of Engineering,Pune*
[5] *Mayuresh Dhadawale ,Electrical Department & Zeal College of Engineering,Pune*

---***---

Abstract:- In this project we will learn to use an Arduino and an attached charging circuit to control the charging of NiMH rechargeable batteries. Rechargeable batteries are a great way to power your portable electronics. They can save a lot of money when properly recycled. But it's much more fun to build one for yourself. So, here is how to build an Arduino controlled battery charger. Each type of battery uses a different chemical process to make it work. As a result, each type of battery needs to be charged differently. We are going to focus on the most common type of AA rechargeable battery, Nickel-Metal Hydride (NiMH). Battery Charger is designed for charging 12V sealed lead-acid batteries. The designed device consists Charging unit , Battery Housing Unit (Drawers) with their respective batteries insider the Drawers which can be charged simultaneously. Each Battery Housing Units provided with its driver circuit, transformer and power supply module. Power supply module is designed with thermal analysis & optimization and protection for EMI/EMC issues. Once the battery is connected to the circuit, it then displays battery charging condition. Battery charging level is displayed by LEDS and LCD is used to indicate the keyed input battery voltage and current through keypad 4X4 manually. Battery type and remaining charging time are displayed on screen during charging on LCD display. PIC 18f452 Microcontroller continuously monitors the battery condition and displays it on LCD. Charging stops when battery is fully charged, audio alarm is indicated with a buzzer and finally the ejection of the drawer tray for removal of the charged battery. This is advantageous as it prevents the battery from damage and over charging.

1.Introduction:-

The Battery Charger circuit presented here can charge a 12V, 7Ah battery, or above. Special features of the charger are as follows. It controls the charging current as per the status of the battery. Battery voltage level as well as charging status are indicated on the LCD display. The charger maintains float voltage, if battery is fully charged. Arduino identifies status of the battery connection and voltage, and indicates the same on the LCD.

2. Arduino Controlled Battery Charge:-

Arduino-based control would typically involve researching existing studies, articles, and projects related to Arduino-based battery chargers: Look for projects or research papers that utilize Arduino microcontrollers for battery charging applications. This could include circuit designs, control algorithms, and performance evaluations. charging algorithms: Investigate different algorithms and methods used for charging batteries. This could include techniques such as constant voltage, constant current, pulse charging, and smart charging algorithms. Battery chemistry and charging characteristics: Understand the specific requirements and characteristics of the type of battery you are charging (e.g., lead-acid, lithium-ion, NiMH). This includes factors like voltage thresholds, charging rates, and safety considerations. Efficiency and performance optimization: Explore strategies for optimizing the efficiency and performance of battery chargers, such as power factor correction, temperature monitoring, and adaptive charging algorithms.Safety considerations: Review literature on safety features and considerations for battery charging systems, including overcharge protection, short-circuit protection, and thermal management.User interface and monitoring: Look for studies or projects that include user interfaces for monitoring and controlling the charging process, such as LCD displays, LED indicators, and smartphone integration. Environmental impact: Investigate research related to the environmental impact of battery charging systems, including energy consumption, lifecycle analysis, and end-of-life disposal considerations. By conducting a thorough literature review on these topics, you can gain valuable insights into best practices, design considerations, and potential areas for innovation in developing an battery charger with Arduino-based control.

Define objectives and requirements: Clearly outline the objectives of the project and identify the specific requirements for the battery charger, including the type of battery to be charged, charging voltage and current levels, safety features, and user interface requirements. Research and literature review: Conduct a literature review to gather information on existing battery charging methods, Arduino-based control systems, and relevant safety standards and guidelines. Component selection: Select appropriate components for the battery charger, including the Arduino microcontroller, power supply, charging circuitry (such as voltage regulators and current sensing components), display interface (if applicable), and safety features (such as overcharge protection circuits).Circuit design: Design the circuitry for the battery charger, including the charging algorithm, voltage and current sensing circuits, and any additional safety features. Use simulation tools if necessary to verify the design. Arduino programming: Write the Arduino code to implement the automatic charging algorithm, monitor battery parameters, and control the charging process. This may involve implementing a PID controller or other control algorithms to regulate voltage and current levels during charging.Prototype construction: Build a prototype of the battery charger circuitry and Arduino control system, following the design specifications and component selection. Test the prototype to verify its functionality and performance.Evaluation and testing: Evaluate the performance of the battery charger prototype through testing and validation. This may involve charging different types of batteries, measuring charging efficiency and accuracy, and assessing safety features.Optimization and refinement: Identify any areas for improvement in the design or performance of the battery charger prototype, and make necessary adjustments to optimize its functionality and reliability.Documentation and reporting: Document the design process, including circuit diagrams, Arduino code, test results, and any design challenges orlessonslearned. Prepare a comprehensive report or documentation package to communicate the methodology and findings of the project.By following these steps, you can develop an effective methodology for designing an automatic battery charger with Arduino-based control, ensuring that the final product meets the specified objectives and requirements.Below is a basic outline of the Arduino code for an automatic battery charger. This code assumes that you have already implemented the necessary hardware components such as voltage and current sensing, as well as safety features like overcharge protection. It also assumes you're charging a lead-acid battery, but you can adapt it for other battery .

Block Diagram:-

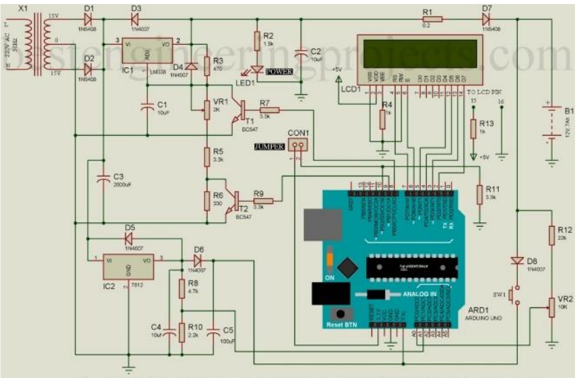

Figure 1: Circuit Diagram of Arduino Controlled 12V, 7Ah Battery Charger

An automatic battery charger with Arduino typically works by monitoring the battery voltage and adjusting the charging current accordingly. Here's a basic overview of the working principle

1*Voltage Sensing:*The Arduino measures the voltage of the connected battery using an analog input pin

.2. *Control Logic:* Based on the measured voltage, the Arduino determines the charging stage (e.g., constant current, constant voltage, trickle charge).

3. *Charging Current Control:* The Arduino adjusts the charging current through a charging circuit (e.g., using a MOSFET) to regulate the charging process.

4. *Feedback Mechanism:* Continuous monitoring ensures that the charging voltage and current stay within safe limits. If necessary, the charging parameters are adjusted dynamically.

5. *Display and User Interface:* An optional LCD display or other interface may be used to show charging status, battery voltage, and other relevant information.

6. *Safety Features:* Overcharge protection, over-discharge protection, and temperature monitoring are often implemented to ensure safe charging.

7. *Timer and Trickle Charge:* The Arduino can control the charging duration and initiate a trickle charge once the battery is fully charged to prevent overcharging.

8. *Power Source Selection:* Some designs allow switching between different power sources (e.g., solar, AC) for versatile charging.

1) RESISTOR:-Resistance is used to the flow of current.Whean resistor is placed in a circuit ,the current flow decreses when current passes through thr resistor .The part of currnt energh dissipate in the from of heat in resistor ,thus decreses the total current.

2) CAPACITOR:-A Capacitors are used in circuits to store and manage electrical charge, smoothing voltage fluctuations, filtering signals, and various other applications in electronic devices.

3) DIODE:- Diodes are fundamental components in electronics, used for signal demodulation, voltage regulation, and protection against reverse voltage.

4) TRANSISTOR:-A transistor is a semiconductor device that can amplify or switch electronic signals and electrical power.

5) IC :-An Integrated Circuit (IC) is a miniaturized electronic circuit consisting of multiple interconnected semiconductor devices, such as transistors, resistors, capacitors, and diodes, on a single semiconductor substrate.

6)TRANSFORMER:-A center-tapped transformer is a type of transformer with a tapped winding at its center. This means that the secondary winding has a midpoint connection, creating two equal halves. The primary purpose of a center-tapped transformer is to provide a split or dual-voltage output.

7) ARDUINO :-The Arduino Uno is a microcontroller board based on the ATmega328P microcontroller.

8) JUMPER WIRE :-Jumper wires are electrical wires with connector pins at each end.

9) BATTERY :-A battery stores and releases electrical energy through a chemical reaction. When connected to a device, it supplies power by converting chemical energy into electrical energy, allowing the device to operate. Recharging reverses this process, restoring the chemical potential within the battery for future use.

3. CONCLUSIONS

The conclusion for an Arduino controlled battery charger is that it can be a useful and safe option for battery management. It can provide excellent technical capabilities and safety features, helping prevent damage due to overuse of the battery. Additionally, an Arduino controlled battery charger can offer versatility and the ability to customize, allowing users to work with various types of batteries.

ACKNOWLEDGEMENT:-

I understand you're looking for information or guidance on an Arduino-controlled battery charger. Arduino can be a powerful platform for creating customized battery charging solutions. Could you please provide more details about what specifically you need help with or what information you're seeking? This will help me provide a more targeted response.

REFERENCES

https://www.google.com
https://openai.com/blog/chatgpt
https://www.arduinocloud.com
https://www.youtube.com