International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04580
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Battery Monitoring System (BMS) Using Arduino UNO

Banoth Srinu[1], Mahesh Mudavath[1*], G.Anil[2], M.Sai Kumar[3], T.Naresh[4], P.Varshith[5]

ECE Department, Siddhartha Institute of Technology & Sciences, Narapally, Ghatkesar, Medchal-Malkajgiri-50008, India.
Banothsrinu_cse@siddhartha.co.in , maheshmudavath.ece@siddhartha.co.in, 223tq1a0407@siddhartha.co.in, 23tq1a0411@siddhartha.co.in
23tq5a0423@siddhartha.co.in , 23tq5a0458@siddhartha.co.in

----------***----------

Abstract - A Battery Monitoring System (BMS) is an electronic setup designed to track essential parameters of rechargeable batteries, such as voltage, current, and State-of-Charge (SoC). By preventing overcharging and over-discharging, such systems help extend the lifespan and reliability of batteries. However, commercially available BMoS solutions are often costly and unsuitable for budget-friendly embedded systems.

Given the widespread use of Arduino Uno for its affordability, open-source platform, and user-friendly programming environment, this study aims to develop a BMoS using an Arduino Uno microcontroller. The proposed system includes voltage and current sensors, an Arduino Uno board, and a liquid crystal display (LCD) for real-time monitoring.

To achieve this, the study set out three primary objectives. First, it was necessary to mathematically establish the relationship between the sensors' input and output values. These mathematical expressions were then validated by observing sensor outputs under varying load conditions — by connecting and disconnecting the load and monitoring the corresponding readings.

Following this, a complete BMoS prototype was assembled by integrating the sensors and LCD with the Arduino Uno. The system was tested using an 11.1 V Lithium-ion battery connected to a DC motor as a load. Test results showed that the current sensor read zero when no load was present, confirming no current flow. Additionally, the LCD accurately displayed a battery voltage of 11.1 V when fully charged.

This developed BMoS allows users to continuously monitor a battery's voltage, current, and SoC, ensuring it is neither overcharged nor excessively discharged. The system provides a practical, low-cost solution for managing battery performance in various electronic applications. Ultimately, the prototype demonstrated its effectiveness by offering users a simple, efficient means of supervising battery operation, thereby enhancing system reliability and ease of use.

***Key Words*:** Battery Monitoring System, State-of-Charge, liquid crystal display, Arduino Uno, current sensors.

I. Introduction:

Batteries play a vital role in modern life, serving as essential energy storage solutions across a wide range of applications — from portable consumer electronics and electric vehicles to renewable energy systems. As the world shifts toward cleaner energy and electrified transport, the demand for reliable, efficient batteries continues to grow at an unprecedented pace.

Incorporating battery energy storage systems offers numerous advantages, including improved integration of renewable energy sources into the grid, reduced dependency on conventional power supplies, backup power availability, lower carbon emissions, and long-term financial savings. Industrial batteries, in particular, are designed with an emphasis on durability, reliability, and performance. Depending on specific application requirements, different battery types — such as Lithium-ion, Lead-acid, and Nickel-Cadmium — are chosen based on factors like cost, weight, size, and maintenance needs [1-5].

Despite their benefits, batteries are susceptible to several operational issues, including overcharging, voltage instability, overheating, and gradual capacity loss. These problems not only affect battery performance but also pose safety risks and may eventually lead to system failure. To address these challenges, battery monitoring systems have been developed to continuously track critical parameters such as temperature, voltage, current, and state of charge (SOC). Real-time monitoring enables optimization of charging and discharging cycles, helps prevent damage from abnormal conditions, and extends battery lifespan [6-8].

In addition to monitoring, battery protective systems are designed to activate safety mechanisms when any parameter crosses predefined safety thresholds. These systems issue early warnings for potential issues like low SOC or excessive temperatures, allowing operators to take timely corrective action and avoid damage.

A key enhancement in modern battery management is the integration of remote monitoring and control capabilities. This feature allows operators to supervise battery health and performance from a distance, enabling predictive maintenance and minimizing operational downtime.

In this project, a battery monitoring and protective system is developed to continuously oversee crucial battery parameters, including voltage, current, temperature, and SOC, utilizing an Arduino-based setup. The system employs a combination of sensors, control circuits, and smart algorithms to collect and analyze real-time data. The BLYNK app is used as the

interface for remote data visualization and management. Based on the collected information, appropriate protective actions are automatically triggered to safeguard the battery and connected equipment.

II. LITERATURESURVEY

Maltezo et al. [1] introduced a battery management system (BMS) specifically designed for lead-acid battery banks in electric vehicles. This system integrates diagnostic, measurement, and monitoring functionalities aimed at enhancing the performance, efficiency, and conservation of lead-acid batteries. It precisely tracks various parameters such as no-load voltage, load voltage, mean and maximum current, maximum temperature, depth of discharge (DoD), charge-discharge cycles, and determines the battery's end-of-life. The study found that when the discharge current remains low, the depth of discharge has minimal impact on the overall battery lifespan.

In another study, Shivanand Basgonda Patil et al. [2] emphasized the importance of continuously providing users with the overall health status of the battery pack, alongside monitoring essential parameters such as voltage, current, temperature, and other key metrics. The primary focus of this work was on the design and implementation of State of Charge (SoC) and State of Health (SoH) estimations for lead-acid batteries. An Arduino-based measurement system was employed, functioning as a voltmeter, ammeter, ohmmeter, and thermometer. The diagnostic component collected data from an SD card module, battery specifications, and sensor readings, which were subsequently stored in the system's database. The real-time monitoring interface displayed critical parameters, SoC, and estimated remaining battery operation time in hours and minutes. Experiments were conducted to assess the State of Health and Depth of Discharge using BattMan 3M1 and 3M2 devices.

Similarly, Nurul Fitriyah Roslan et al. [3] focused on developing a Battery Monitoring System (BMoS) for rechargeable battery packs, capable of tracking parameters like battery voltage, current, and SoC to prevent overcharging and over-discharging, thereby extending battery life. The system employed mathematical equations within its Arduino Uno programming to convert 10-bit ADC readings into actual voltage and current values measured by the sensors. The measured current value was then used to compute the SoC. A working prototype was tested with an 11.1 V Lithium-ion battery powering a DC motor, and the battery's discharging process, SoC, and voltage values were recorded and displayed on an LCD.

Niraj Agarwal et al. [4] presented a smart energy management system for lead-acid batteries, developed and tested using Matlab and Arduino platforms. The system incorporated an ACS712 sensor for current and voltage measurements, and an LM35 thermistor for temperature detection. Sensor data was processed and stored through an Arduino microcontroller. This paper also explored various techniques for estimating the State of Charge (SoC), including direct measurement, book-keeping estimation, adaptive systems, and hybrid methods. In their approach, a hybrid method was applied — using the book-keeping estimation based on open-circuit voltage and temperature for static SoC, and direct estimation through charge flow monitoring for dynamic SoC. The study highlighted the significant effect of temperature on the SoC and stressed the importance of developing BMS solutions that accurately account for the relationship between temperature and battery charge levels.

III. PROPOSED SYSTEM

A. Objectives of the Proposed Model

The primary goals of the proposed Battery Monitoring and Protective System are outlined as follows:

1. **Real-Time Battery Health Monitoring:**

   The system is designed to continuously observe the battery's condition and performance. It measures key parameters such as voltage, current, temperature, and State of Charge (SoC), ensuring that the battery operates safely and within optimal ranges [5], [6].

2. **Protection Against Damage:**

   One of the critical functions of the system is to safeguard the battery from risks like overcharging, over-discharging, overheating, and short-circuiting. The protective mechanism activates when any monitored parameter surpasses preset safety thresholds, thereby preventing potential damage.

3. **Optimizing Performance and Efficiency:**

   The monitoring system aims to improve battery efficiency by regulating the charging and discharging processes. This ensures the battery charges only when necessary and discharges as required, helping to enhance both its operational lifespan and overall efficiency [5], [6,10].

4. **Providing Early Alerts:**

   The system can detect abnormal battery conditions, such as critically low SoC or elevated temperatures, and deliver early warnings. This allows timely corrective

actions, preventing operational issues and prolonging battery life.

5. **Enabling Remote Monitoring and Control:**

   An important feature of the system is its capability for remote supervision and control. It supports predictive maintenance by allowing operators to access real-time data on battery health and performance from remote locations, thereby reducing system downtime [7], [8-15].

In summary, the proposed Battery Monitoring and Protective System is designed to ensure the safe, efficient, and reliable operation of battery-powered systems, extend battery service life, and reduce the risk of unexpected failures.

B. Hardware Requirements

The block diagram of the proposed Battery Monitoring and Protective System is depicted in **Fig. 1**. The main hardware components required for its implementation include:

- **Sensors (Temperature, Current, Voltage):**

  These are essential for continuously measuring the critical parameters of the battery. The sensors are connected in series with the battery and in parallel with the data acquisition and monitoring units, including an LED display, Arduino Uno, and other peripheral devices.

- **Arduino Uno Microcontroller:**

  It serves as the central processing unit of the system, receiving data from the sensors. Based on predefined instructions, it calculates the State of Charge (SoC) and processes other vital parameters [17-20].

- **LCD Display:**

  The calculated SoC and other real-time battery readings are displayed on the LCD screen for immediate monitoring.

- **Charging and Discharging Units:**

  The battery discharges while supplying current to a DC motor, and the charging process is facilitated by connecting the battery to a charger. The system monitors and controls both processes to ensure safe and efficient operation.

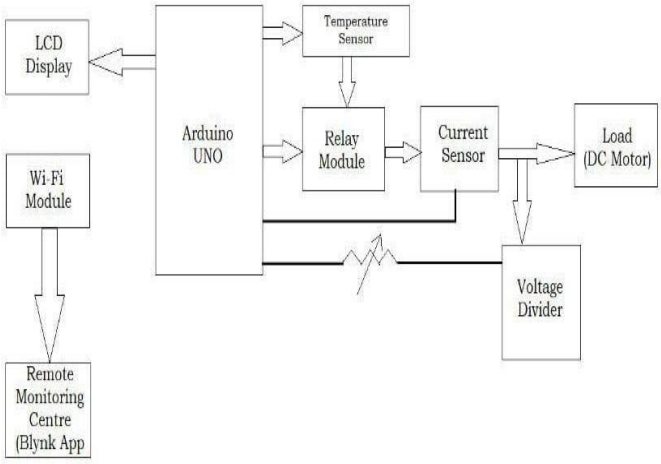


Fig.1BlockDiagramofProposedBatteryMonitoringandProtectionSystem

IV. IMPLEMENTATION

The current, voltage, and temperature sensors were individually tested, and their outputs were successfully recorded. During this process, the Arduino was programmed to interface with each sensor, ensuring proper functionality. Additionally, sensor calibration was performed to enhance accuracy. The connection layout for the temperature sensor with the Arduino is illustrated in **Fig. 2**.

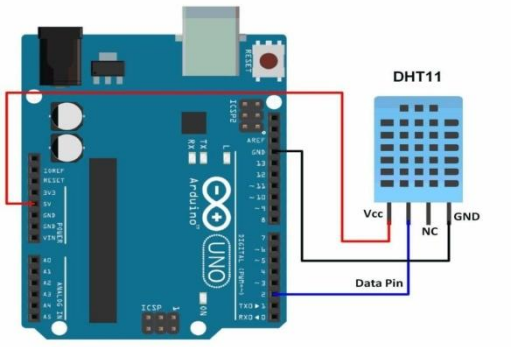


Fig.2:Circuit diagram of temperature sensor with Arduino

A. Current Sensor Algorithm

1. Set up the sensor input pin as **A0** and define a constant for **voltage per ampere**.
2. Initiate **serial communication** at a baud rate of **9600** within the setup() function.
3. Declare and initialize variables to store **voltage**, **RMS voltage**, and **RMS current** values.
4. In the loop() function, obtain the **peak-to-peak voltage** using the getVPP() function.
5. Compute the **RMS voltage** by dividing the peak-to-peak voltage by **2√2**.
6. Determine the **RMS current** by dividing the RMS voltage by the **voltage per amp constant**.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04580
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

7. Display the calculated **RMS current value (in Amps)** on the **serial monitor [21-25]**.

B. Voltage Sensor Algorithm

1. Configure the voltage sensor's input pin as **A0** and define a constant for **voltage per ampere**.
2. Begin **serial communication** at **9600 baud rate** inside the setup() function.
3. Declare variables for **peak-to-peak voltage**, **RMS voltage**, and **RMS current**.
4. In the loop() function, acquire the **peak-to-peak voltage** using the getVPP() method.
5. Calculate the **RMS voltage** from the peak-to-peak reading using the **√2 factor**.
6. Output the resulting **RMS voltage value (in Volts)** to the **serial monitor**.

C. Temperature and Humidity Sensor Algorithm

1. Include the **SimpleDHT library** and assign a pin number for connecting the **DHT11 sensor**.
2. Within the setup() function, initialize **serial communication** and print a header such as **"Temperature and Humidity Data."**
3. In the loop() function, read **temperature and humidity values** from the DHT11 sensor using the read() function provided by the SimpleDHT library.
4. If the reading is successful, display the **temperature in °C** and **humidity as a percentage** on the serial monitor.
5. Introduce a **delay of 1.5 seconds** before the next measurement cycle begins.

V. RESULTANDDISCUSSION

A. Current Sensor Connection and Output

The current sensor setup was tested both with and without a lamp connected as a load, as illustrated in **Fig. 5**. For measuring the current, an **ACS712 current sensor** was employed in the circuit. The Arduino's serial monitor displayed the sensor's output under three conditions: **no load**, **higher load**, and **lower load**, with the corresponding results shown in **Fig. 6**.

B. Voltage Sensor Connection and Output

The connections for the **ACS712 voltage sensor** are illustrated in **Fig. 8**, configured to monitor the system's normal operating voltage. As shown in **Fig. 9**, a **voltage divider circuit** is implemented to enable accurate voltage measurement. In line with the voltage sensor algorithm, the system is programmed to display the voltage value detected by the sensor. If the battery voltage drops below a predefined threshold, a **relay is activated to disconnect the circuit**, preventing further discharge. This mechanism also helps estimate the remaining energy in the battery. By determining the **State of Charge (SoC)**, the system can predict when the battery may require maintenance or replacement, reducing the risk of unexpected system downtime or failures.

Fig. 3: Current sensor with Arduino for Higher Load

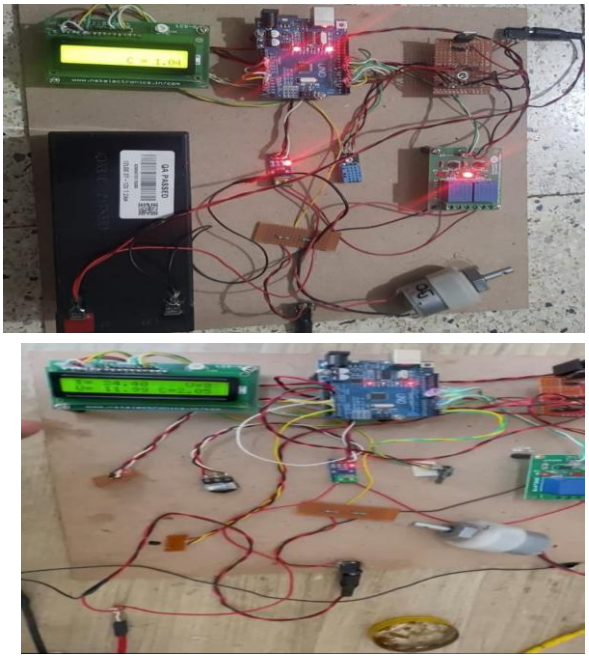
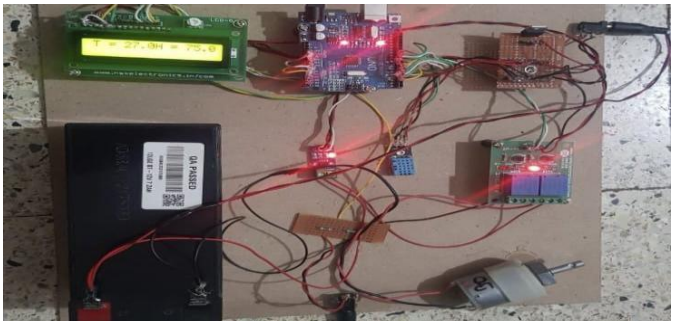


Fig.4:Current sensor connection with battery



CONCLUSION

The proposed battery monitoring system incorporates a **temperature sensor, current sensor, and voltage divider** to measure the battery's temperature, current, and voltage, all interfaced through an **Arduino**. With appropriate programming, the operation of these sensors, along with the load, battery, and relay, was successfully observed. Initially, the setup was tested using an **AClamp load** for analysis, followed by actual monitoring and protection with a **Lithium-ion battery**. In the final stage, **real-time measurements** of the key parameters — voltage, current, and temperature — were performed directly on the battery. Additionally, the system is designed to automatically **disconnect the circuit if the voltage crosses a predefined limit** to safeguard the battery.

The **BLYNK application** was integrated into the setup, enabling wireless observation of these battery parameters on a mobile device. This kind of battery monitoring system, when embedded in **portable electronic devices**, helps protect against hazards such as **overcharging, over-discharging, and short circuits**. Furthermore, the model can be extended to display the **State of Charge (SoC)**, assisting in optimizing battery life by preventing excessive charging or deep discharging — both of which can degrade battery performance or cause irreversible damage. Beyond ensuring safe operation, this system also supports **remote monitoring and predictive maintenance**, reducing potential downtime and minimizing risks like **device malfunctions or battery-related accidents**.

REFERENCES

1. Maltezo, M. R. C., Thio-Ac, A. C., Castillo, A. M. C., Gattu, L. E., Hernandez, C. E. A., Labuan, J. J. C., Navales, L. F., Sopeña, E. C., Arago, N. M., Galido, E. A., Madrigal, G. A. M., Pascion, C. G., & Tolentino, L. K. S. (2021). *Arduino-based battery monitoring system with state of charge and remaining useful time estimation*. International Journal of Advanced Technology and Engineering Exploration, 8(76), March 2021.A. Kumar and R. Patel, "IoT Based Smart Surveillance System Using Raspberry Pi," *International Journal of Engineering Research & Technology*, vol. 9, no. 5, 2020.
2. Shivanand Basgonda Patil, Akshay Sanjay Kamme, Aniket Sunil Patil, Vaibhav Ravso Patil, Apurva A Londhe. (2020). *Design of Battery Health Monitoring System Using Arduino Uno*. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE), 9(6), e-ISSN: 2278-8875, p-ISSN: 2320-3765. Impact Factor: 7.122.Raspberry Pi Foundation, "Raspberry Pi 4 Model B," [Online]. Available: https://www.raspberrypi.org
3. Nurul Fitriyah Roslan, & Wan Mariam Wan Muda. (2020). *Development of Battery Monitoring System Using Arduino Uno Microcontroller*. University Malaysia Terengganu Journal of Undergraduate Research, 2(4), 41-50. eISSN: 2637-1138.
4. Niraj Agarwal, Phulchand Saraswati, Ashish Malik, & Yogesh Krishan Bhateshvar. (2017). *Design a Battery Monitoring System for Lead-Acid Battery*. International Conference Proceeding ICCCT, Dec 2017, ISSN: 2320-2882, RIET Jaipur & IJCRT.ORG.
5. Jun Lee, Xiaosong Hu, Xueliang Huang, & Shulong Zhou. (2017). *A Review of Battery Monitoring and Management Systems for Electric Vehicle Applications*. Energies.
6. Omar Hegazy, Fatma Ashour, & Noha A. El-Zaher. (2015). *Battery Management Systems in Electric and Hybrid Vehicles: A Review*. Journal of Power Sources.
7. Z. Yang, D. Xiong, X. Hu, Z. Guo, & L. Zheng. (2014). *Battery Monitoring System Based on Wireless Sensor Networks for Electric Vehicles*. IEEE Transactions on Industrial Informatics.
8. Weiming Wang, Xiaoqiang Guo, & Kezhao Pang. (2018). *Battery Management Systems in Electric Vehicles*. IEEE Transactions on Vehicular Technology.
9. **Mahesh Mudavath**, K. Hari Kishore, Azham Hussain and C.S. Boopathi "Design and analysis of CMOS RF receiver front-end of LNA for wireless applications", in "*Microprocessors and Microsystems*, Volume 75, 13th January. **2020**, 102999 (SCI+ Scopus+ web of science)", doi: https://doi.org/10.1016/j.micpro.2020.102999.
10. Thurpati, S., **Mudavath, Mahesh**. & Muthuchidambaranathan, P. "Design of Hybrid Beamforming for Multiuser MIMO mmWave Systems Using Deep Learning", *Wireless Personal Communication* (**2024**), https://doi.org/10.1007/s11277-024-11159-3
11. Sammaiah Thurpati, **Mahesh Mudavath,** S.Sandhya Rani, G. Koteswara Rao, P Muthuchidambaranathan" Enhanced User Clustering Optimization Based on HBF for mmWave Massive MIMO Systems" Wireless Personal Communication (**2024**), Springer, (SCI+ Scopus+ web of science). https://doi.org/10.21203/rs.3.rs-2826210/v1
12. **Mahesh Mudavath**, S.Sandhya Rani, Dubbaka Shirisha, and Banoth Srinu "A Two-Stage Method for Hybrid Precoding in Downlink Millimeter Wave Massive MIMO", in Journal of Engineering Science and Technology Review, Volume 18, Issue 2, **2025**, page: 140-148, ISSN: 1791-2377, DOI: 10.25103/jestr.182.18
13. Sammaiah Thurpati, **Mahesh Mudavath**, S.Sandhya Rani "Human Behavior Recognition System Based on Low-Cost IoT Chip ESP32", in Journal of Engineering Science and Technology Review, Volume 18, Issue 2, **2025**, Page: 178-185, ISSN: 1791-2377, DOI: 10.25103/jestr.182.22
14. **Mahesh Mudavath**, Banoth Srinu, Murali Anumothu, and V.Sabitha "Hybrid Precoding methods for mMIMO Vehicle-to-Vehicle System", in Journal of Engineering Science and Technology Review, Volume 18, Issue 3, **2025**, ISSN: 1791-2377, DOI: 10.25103/jestr.182.18
15. **Mahesh Mudavath**, Sresta Valasa, Avunoori Saisrinithya and Amgothu Laxmi Divya "Design of Cryogenic CMOS LNAs for Space Communications", in Journal of Physics: Conference Series, Volume 1817, Issue 1, **2021**, 012007, ISSN: 1742-6588, E-ISSN: 1742-6596, IOP Publishing, DOI: 10.1088/1742-6596/1817/1/012007
16. **Mahesh Mudavath**, Amgothu Laxmi Divya, Ch S Ranadheer, Mohamed Afzal and R.Venkateswarlu "Low Noise Amplifier Design and Performance Analysis of RF Front-End for Narrow

Band Receivers", in Journal of Physics: Conference Series, Volume 1817, Issue 1, **2021**, 012005, ISSN: 1742-6588, E-ISSN: 1742-6596, IOP Publishing DOI:10.1088/1742-6596/1817/1/012005

17. Thurpati, S., **Mahesh Mudavath.**, Muthuchidambaranathan, P. "Performance analysis of linear precoding in downlink based on polynomial expansion on massive MIMO systems" in Journal of Physics: Conference Series, Volume 2062, Issue 1, **2021**, 012006, ISSN: 1742-6588, E-ISSN: 1742-6596, IOP Publishing, DOI: doi:10.1088/1742-6596/2062/1/012006,
18. Sresta Valasa, D.R.Ramji, Jitesh Shinde and **Mahesh Mudavath**, "An improved impulse noise removal VLSI architecture using DTBDM method", in Springer Lecture Notes on Data Engineering and Communication Technologies, Vol. 63, pp: 325-335, Series Ed.:Xhafa, Fatos, ISSN: 2367-4512, **2021**, DOI: 10.1007/978-981-16-0081-4_32
19. **Mahesh Mudvath**, K.Hari Kishore "A Low NF, High Gain of 2.4GHz differential LNA design for wireless applications" in *International Journal of Scientific & Technology Research (IJSTR), VOL. 8, ISSUE 12,* pp: 109-116, December-2019, ISSN:2277-8616
20. **Mahesh Mudavath** and K. Hari Kishore "*RF Front-End Design of Inductorless CMOS LNA circuit with noise cancellation method for IoT Applications"* in *International Journal of Innovative Technology and Exploring Engineering (IJITEE)" Vol.8, Issue-6S, page: 176- 183, 2019,* ISSN: 2278-3075
21. **Mahesh Mudavath** and K. Hari Kishore "Low Noise Amplifier Design of CMOS Inductorless with Noise Cancellation Technique" in *International Journal of Engineering and Advanced Technology (IJEAT)*, ISSN: 2249–8958, Volume-9, Issue-1, October 2019, pages1537-1542
22. **Mahesh Mudavath** and K. Hari Kishore "*Design and Analysis of Receiver Front-End of CMOS cascode common source stage with inductive degeneration low noise amplifier on 65nm technology process" Journal of computational and theoretical nanoscience (JCTN)" Vol.16, No.5-6, page: 2628-2634, May-2019,* ISSN: 1546-1955 (Print); ISSN 1546- 1963(Online)
23. **Mahesh Mudavath** and K. Hari Kishore "Differential CMOS Low Noise Amplifier Design for Wireless Receivers" in International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, Volume-8, Issue-4, 2019, pp: 2467-2474
24. **Mahesh Mudavath** and K. Hari Kishore "CMOS Front-End of Low Noise Amplifier for GPS and GSM wireless Applications" in *International Journal of Engineering and Technology (UAE)*, 2018, Vol. No. 7(1.5), ISSN: 2227-524X, pages 1-6. DOI: 10.14419/ijet.v7i1.5.9069
25. **Mahesh Mudavath** and K. Hari Kishore "Design of RF Front-End CMOS Cascade CS Low Noise Amplifier on 65nm Technology Process" in *International Journal of Pure and Applied Mathematics*, 2017, Vol. No. 115 and Issue No. 7, ISSN:1314-3395, pages417-422