DESIGN OF A BATTERY CHARGER FOR LITHIUM-ION BATTERY USING BATTERY MANAGEMENT SYSTEM

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY PATNA

APRIL 2025

Report submitted by

Y. THARUN TEJA

2201EE77

ABSTRACT

This project focuses on implementing cell balancing, an important feature of a Battery Management System (BMS) that helps improve the performance and life of battery packs. To do this, we use a smart charging method that carefully controls both current and voltage and a PWM (Pulse-Width Modulation) signal helps control the switching of power to the batteries.

The hardware setup has three batteries connected in series, and an arduino is used as PID controller to keep the charging current accurate. Once the batteries reach a set voltage, the system switches to a constant voltage charging mode.

A flyback converter is used to step-down the input voltage and provide a stable output for charging. The project uses active cell balancing using a delta-structured switched capacitor method, this ensures all the battery cells are isolated and charged separately based on voltage differences.

Introduction:

Overview:

Battery Management Systems (BMS) are critical for the safe and optimal operation of battery packs, especially in electric vehicles (EVs). Early BMS designs provided basic monitoring and protection; modern systems now include advanced features like state-of-charge (SoC) estimation, state-of-health (SoH) monitoring, and cell balancing. These functionalities prevent individual cell overloading, thus enhancing safety and performance.

Types of BMS:

- Centralised BMS: All control is managed by a central unit.
- Distributed BMS: Control functions are spread across multiple smaller units near the cells.
- Integrated BMS: Combines BMS with other vehicle or system management functions for improved efficiency.

Key Functionalities:

- Voltage, current, and temperature monitoring
- Cell balancing
- SoC and SoH estimation
- Cell protection (overcharge/discharge, temperature)
- Charging/discharging control
- Real-time battery data display

What Does a Battery Management System (BMS) Do?

A BMS plays a vital role in keeping a battery pack healthy, safe, and performing well. Here's a breakdown of its key features:

- Cell Monitoring
  The BMS constantly keeps an eye on each individual battery cell—checking things like voltage, temperature, and sometimes current—to make sure everything is running smoothly and safely.
- Voltage & Current Measurement
  It regularly checks the **voltage** of each cell to understand how much charge is left. It also monitors the current going in and out of the battery to ensure it's operating safely and efficiently.
- Temperature Monitoring
  The system tracks the battery's temperature to make sure it's not getting too hot or too cold, since extreme temperatures can damage the battery or cause it to perform poorly.
- Cell Balancing
  Sometimes, some cells in a battery charge or discharge faster than others. Cell balancing makes sure all the cells stay at the same energy level. This helps avoid overcharging or undercharging any one cell, which improves overall performance and extends battery life.
- State of Charge (SoC)
  This tells you how much battery is left, kind of like a fuel gauge. It's calculated using:

$$SoC_t = \frac{Q_t}{Q_n}$$

- State of Health (SoH**)**
  This gives an idea of how "healthy" the battery is overall—how much life it has left, and whether it's performing as well as it did when it was new.
- Cell Protection
  The BMS includes safety features to prevent issues like overcharging, deep discharging, and overheating. This protects both the battery and the user.
- Charging & Discharging Control
  It manages how the battery charges and discharges, making sure the power flow is smooth, efficient, and safe always.
- Battery Display Data
  The BMS can show real-time information such as voltage, current, temperature, SoC, and SoH. This helps users keep track of the battery's status and make smart decisions.

Cell Balancing Techniques:

Cell Balancing:

Ensures all cells in a battery pack have similar voltage and SoC, preventing overcharge/discharge and optimizing performance and lifespan.

Passive Cell Balancing

- Uses resistors to dissipate excess energy from higher-voltage cells as heat.
- Simple, cost-effective, but less energy efficient due to energy loss as heat.

- Example: Switched shunting resistor method, where resistors are switched on when cell voltage exceeds a threshold1.

Active Cell Balancing

- Actively transfers charge between cells using electronic circuits (e.g., DC-DC converters).
- More efficient than passive methods, as energy is redistributed rather than dissipated.

Switched Capacitor Method (Delta Structure)

- Employs capacitors and switches to transfer charge between cells.
- When a cell's voltage is high, a capacitor charges from it and then discharges to a lower-voltage cell, equalizing voltages.
- Delta-structured switched-capacitor equalizers offer fast balancing and high efficiency, independent of cell count or initial voltages.

Flyback Converter Method

- Utilizes a transformer with isolated primary and secondary windings.
- Each cell is connected to a secondary winding via a diode; the primary is connected to the power source.
- When the main switch is activated, energy is stored in the transformer and then transferred to cells as needed.
- Offers high efficiency and straightforward design.
- For this project I have opted to Flyback converter method, so we look into the simulation details;

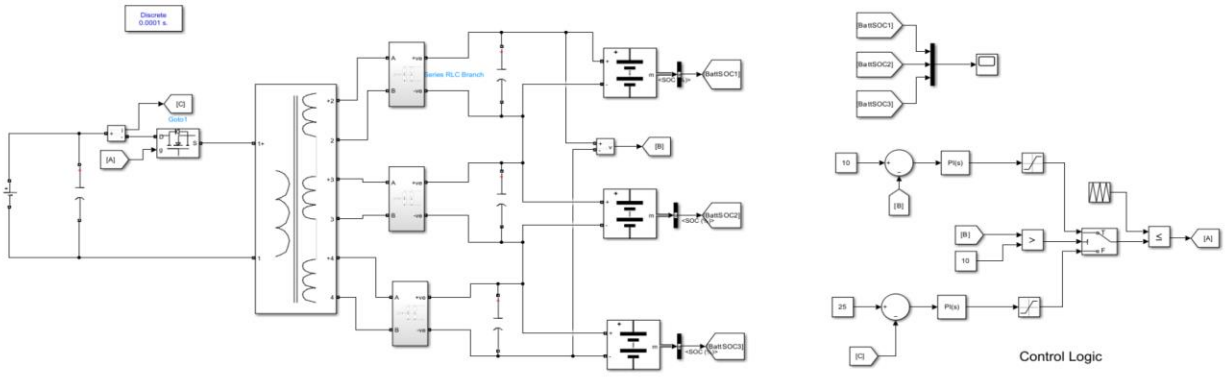

Figure: Simulation of cell balancing circuit using Flyback Converter.

Result:

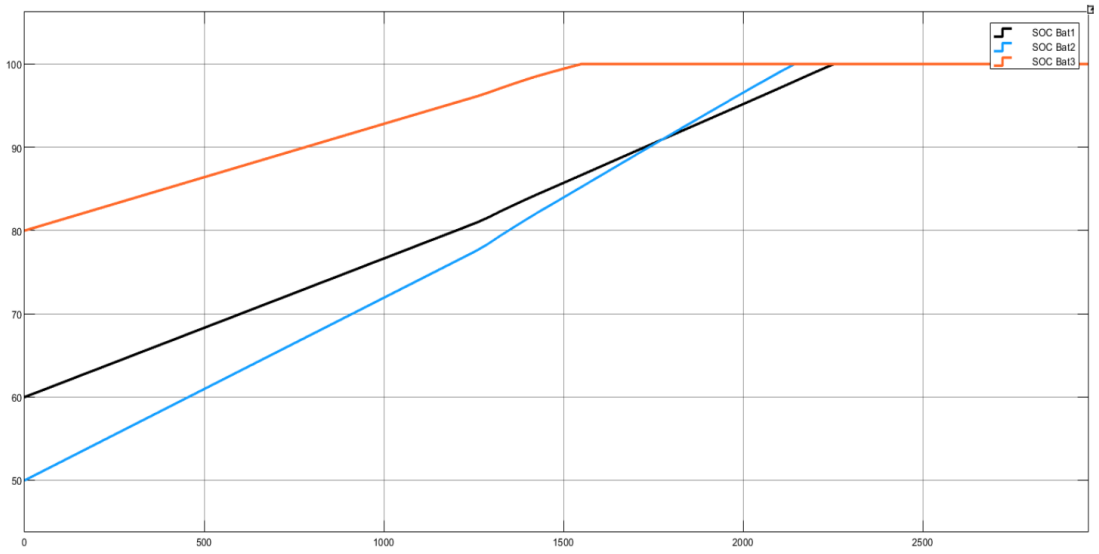

Figure: Simulation result of cell balancing circuit using Flyback Converter.

Hardware Implementation

Diode Bridge Rectifier:

Converts AC to DC using four diodes in a bridge configuration, provides full-wave rectification, meaning it converts both halves of the input AC waveform into a pulsating DC output.

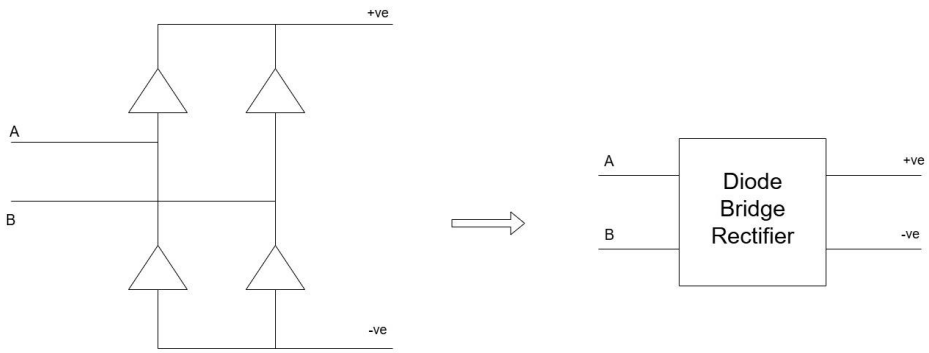

Figure: Diode Bridge Rectifier

Isolation Circuit:

Isolation in power electronics is crucial for safety and reliability. It acts like a protective barrier that keeps high-voltage parts of a system separate from low-voltage, sensitive electronics or even the user. Without isolation, a fault on the high-voltage side could lead to dangerous electric shocks or can damage important components. It also helps prevent ground loops, which are unwanted current paths that can introduce electrical noise and affect how your system works.

There are different ways to achieve isolation, such as:

- Opto-isolation
- Fiber optic isolation
- Pulse transformer isolation

In our project, we're using the TLP250 opto-isolator to create a safe connection between the power circuit and the driver circuit. The isolation circuit is powered separately through a transformer with isolated terminals. This setup ensures that there's no shared ground path, further protecting the system from noise and potential damage.

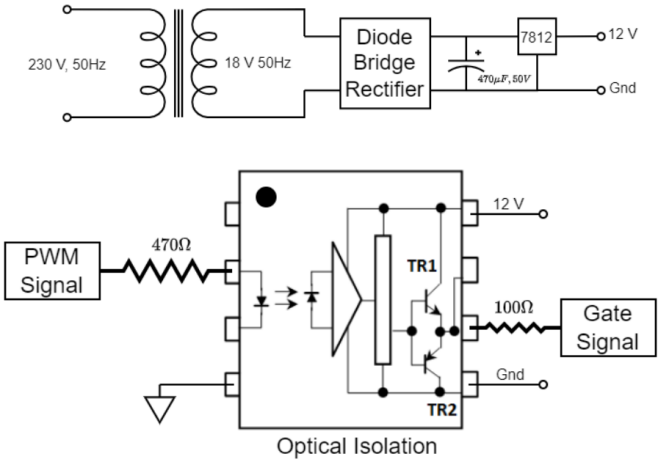

Figure: Isolation Circuit

Sensors:

To monitor electrical parameters like current and voltage in our system, we used two different sensing methods. For current sensing, we used the ACS712-30A sensor, which operates on the Hall effect principle. It provides real-time current measurements and feeds that data to the PID controller for accurate regulation.

For voltage sensing, we used a voltage divider circuit to safely scale down the voltage to a readable level for the microcontroller. Both current and voltage feedback are essential for the PID controller to make precise adjustments, ensure safe operation, and optimize the performance of the system.

Figure: Current sensor

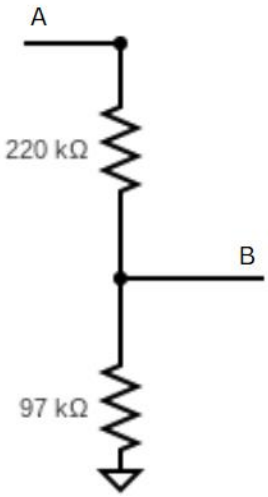

Figure: Voltage Sensor

For voltage sensing, the voltage divider's resistors are choosed using the below formula and I have choose R_1=220kΩ and R_2=97kΩ,

$$V_B = \frac{R_2}{R_1 + R_2} V_A \quad \Rightarrow V_B = \frac{97}{97 + 220} V_A$$

PID Controller and Pulse Width Generation:

For this project, we used the Arduino Uno as the main controller to handle both the PID control and PWM signal generation. The Arduino Uno is a widely used microcontroller board based on the Atmega328P. It offers 14 digital I/O pins, 6 analog inputs, USB connectivity, and a beginner-friendly IDE, making it a great choice for prototyping and control applications.

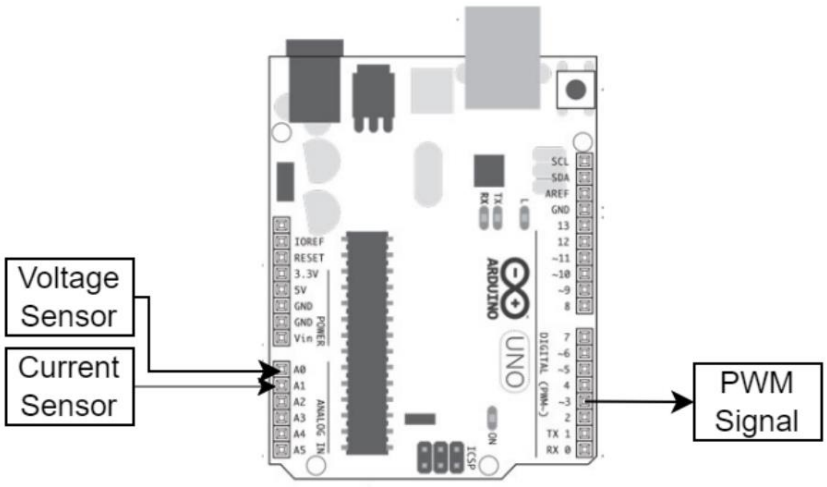

Figure: Arduino UNO

We utilized digital PWM pin 3 of the Uno for controlled pulse generation, operating at a frequency of 25kHz. The pulse width is adjusted dynamically by the PID controller, depending on the feedback from the current and voltage sensors. This allows the system to switch between constant current (CC) and constant voltage (CV) modes during battery charging.

- The voltage sensor output is connected to analog pin A0.
- The current sensor (ACS712-30A) output is connected to **a**nalog pin A1.
- The ACS712 is powered directly from the 5V supply pin of the Uno.

By using the Arduino Uno, we ensured precise control of the charging process while keeping the setup simple and cost-effective.

Multiwinding Transformer:

As discussed above we need a multi winding transformer for creating a cell-balancer in flyback converter. For calculating the primary turns of ferrite core transformer,

Below Table 1 gives the design parameter of transformer, with available $A_{core} = 1.32cm^2$, $B_{max} = 1500\ Gauss$ and frequency of operation is 25kHz.

Table 1: Design parameters of Transformer

V_{in}	20V	V_{out}	15V
I_{in}	10A	I_{out}	5A
$N_{primary}$	10	$N_{secondary}$	7
$AWG_{primary}$	18	$AWG_{secondary}$	20

$$N_{pirmary} = \frac{V_{in} \times 10^8}{4 \times f \times B \times A_{core}}$$

With designed parameters, Table 2 shows the obtained inductance and resistance values of the windings.

Table 2: Inductance and Resistance values of Ferrite Core Transformer.

	Inductance μH	Resistance Ω
Primary	15.482	0.080
Secondary 1	5.514	0.137
Secondary 2	4.622	0.097
Secondary 3	4.773	0.192

Arduino Code Logic:

C++ code is used in Arduino IDE for PID control, for deciding the duty ratio of generated PWM signal for constant current and voltage control. $K_p = 2$, $K_i = 0.9$ and $K_d = 0.5$ are values in PID controller.

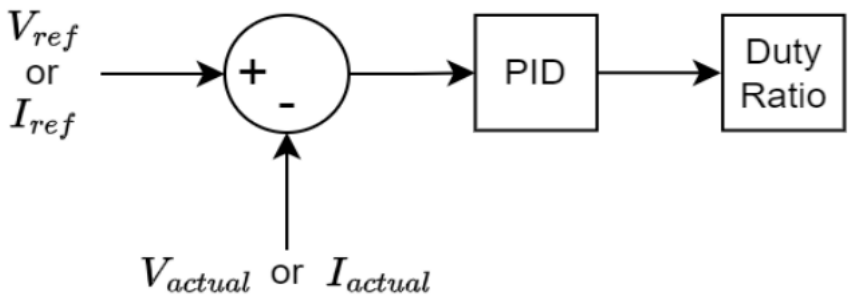

Figure: : PID controller to decide duty ratio of PWM signal using Arduino UNO

For switching the mode of charging from constant current to constant voltage or vice versa, using the if else statement a logic implemented is given below

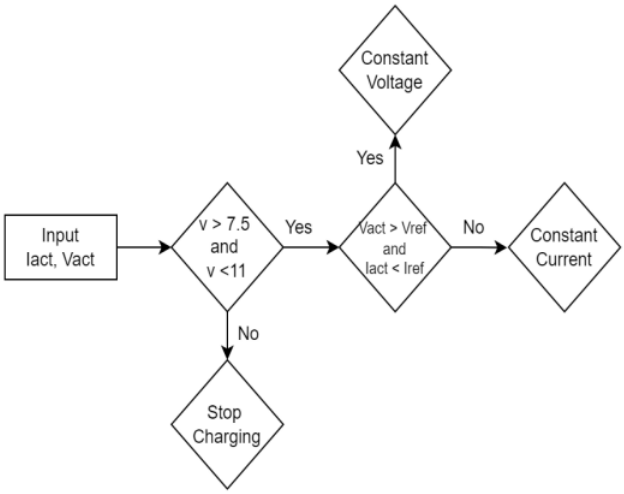

Figure: Logic for switching charging mode

Hardware Set-up:

The complete hardware setup is now assembled to test both cell balancing and battery charging functionalities. Circuit diagram and Hardware setup are also shown below. This design supports cell balancing by delivering isolated, regulated charges to each cell, while enabling current and voltage feedback for precise control using a PID controller (via Arduino).

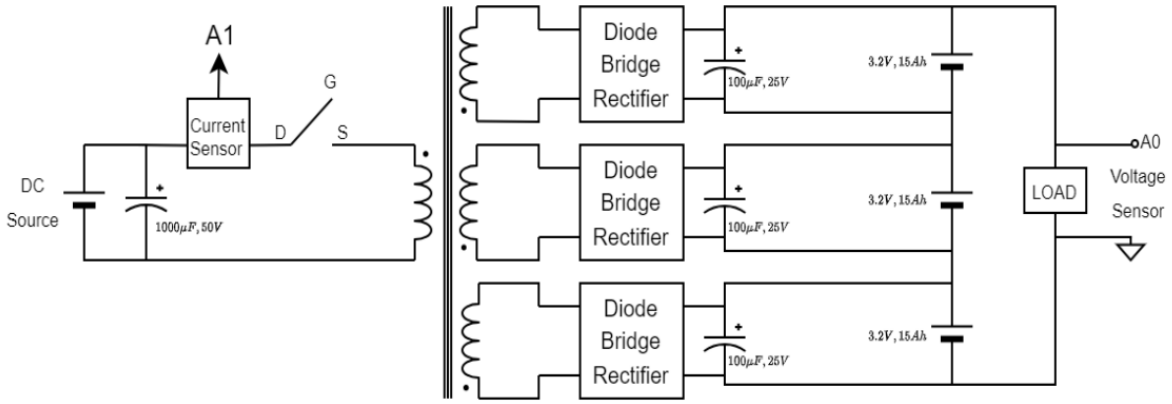

Figure: Topology of cell balancing circuit

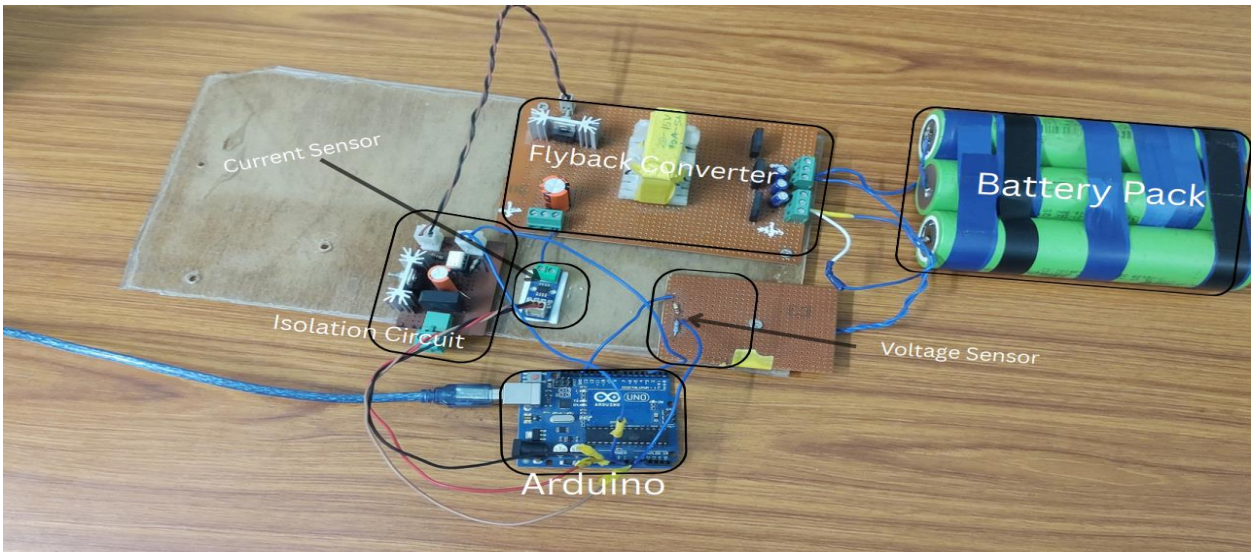

Figure: Hardware of cell balancing circuit

Conclusion:

This report makes a valuable contribution to the advancement of cell balancing techniques in battery management systems. It presents a practical and effective solution through the implementation of a flyback converter-based cell balancing system, which enhances the performance, reliability, and safety of energy storage systems across various applications.

Initially, the project explored active cell balancing using the delta-structured switched capacitor method. However, due to issues such as short circuits and abrupt charging or discharging behavior, the focus shifted to a more stable and efficient flyback converter approach.

Looking ahead, future improvements could involve incorporating bidirectional cell balancing and integrating state of charge and state of health estimation techniques. These additions would further boost the efficiency, intelligence, and dependability of the cell balancing system.

In conclusion, this report not only delivers a robust solution for current challenges but also lays the groundwork for continued innovation in battery management and energy systems.

Arduino Code

```
#include <PWM.h>
#include <PID_v1.h>
#include <LiquidCrystal.h>

// LCD Setup: RS, E, D4, D5, D6, D7
LiquidCrystal lcd(12, 11, 5, 4, 1, 2);

float vOUT = 0.0;
float vIN = 0.0;
float R1 = 221000.0;
float R2 = 95000.0;
float value = 0.0;
float AcsValue = 0.0, Samples = 0.0, AvgAcs = 0.0, AcsValueF = 0.0;
float Iref = 1.5, Vref = 10.0;

int32_t frequency = 25000;

double Kp = 2, Ki = 0.9, Kd = 0.5;
double Input, Output, Setpoint = 0.0;

PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);

void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.clear();

  pinMode(3, OUTPUT);
  InitTimersSafe();
  bool success = SetPinFrequencySafe(3, frequency);
  if (success) {
    pinMode(3, OUTPUT);
    pwmWrite(3, 20);
  }
  myPID.SetMode(AUTOMATIC);
  delay(10000);
}
```

```
void loop() {
  float v, i;
  v = voltage_s();
  i = current_s();
  Serial.println(i);

  display_voltage(v);

  if (v >= 7.5 && v < 11) {
    if (v > 10 && i < Iref) {
      Setpoint = Vref;
      Input = v;
    } else {
      Setpoint = Iref;
      Input = i;
    }

    myPID.Compute();
    pwmWrite(3, Output);
  } else if (v >= 11) {
    pwmWrite(3, 0);
  }
}

float voltage_s() {
  Samples = 0;
  for (int i = 0; i < 150; i++) {
    value = analogRead(A0);
    Samples += value;
  }
  value = Samples / 150.0;
  vOUT = (value * 5.0) / 1024.0;
  vIN = vOUT / (R2 / (R1 + R2));
  return vIN;
}

float current_s() {
  Samples = 0;
  for (int x = 0; x < 150; x++) {
    AcsValue = analogRead(A1);
    Samples += AcsValue;
  }
  AvgAcs = Samples / 150.0;
```

```
  AcsValueF = (2.5 - (AvgAcs * (5.0 / 1024.0))) / 0.066;
  return AcsValueF;
}

void display_voltage(float voltage) {
  lcd.setCursor(0, 0);
  lcd.print("Voltage: ");
  lcd.print(voltage, 2);
  lcd.print(" V   ");
}
```

Arduino References:

- https://docs.arduino.cc/learn/electronics/lcd-displays/
- https://forum.arduino.cc/t/timer1-pwm-25khz/95948
- https://forum.arduino.cc/t/arduino-voltage-divider/636659

References

[1] A. P. Shukla and R. A. Patel, “Battery management system by passive cell balancing for electric vehicle,” in 2022 2nd International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), 2022, pp. 1–6.

[2] M. Nawaz, J. Ahmed, and M. S. Khan, “Cell balancing techniques for li-ion batteries in healthcare devices,” in 2022 Global Conference on Wireless and Optical Technologies (GCWOT), 2022, pp. 1–7.

[3] S. Jeon, J.-J. Yun, and S. Bae, “Active cell balancing circuit for series-connected battery cells,” in 2015 9th International Conference on Power Electronics and ECCE Asia (ICPE-ECCE Asia), 2015, pp. 1182–1187.

[4] Y. Shang, C. Zhang, N. Cui, and C. C. Mi, “A delta-structured switched-capacitor equalizer for series-connected battery strings,” IEEE Transactions on Power Electronics, vol. 34, no. 1, pp. 452–461, 2019.

[5] A. Farzan Moghaddam and A. Van den Bossche, “Flyback converter balancing technique for lithium based batteries,” in 2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST), 2019, pp. 1–4.