Graduation Project Report 1

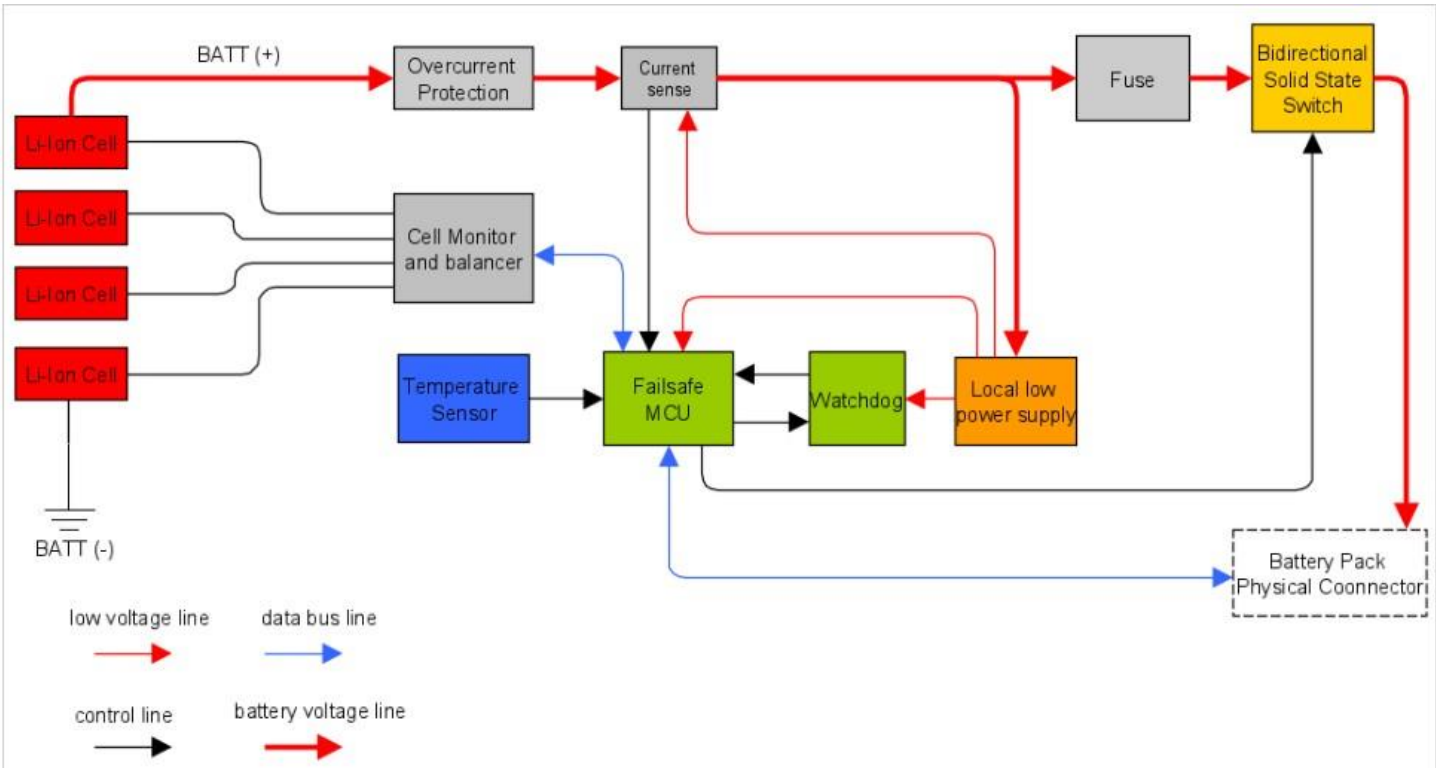
Battery Management System (BMS)

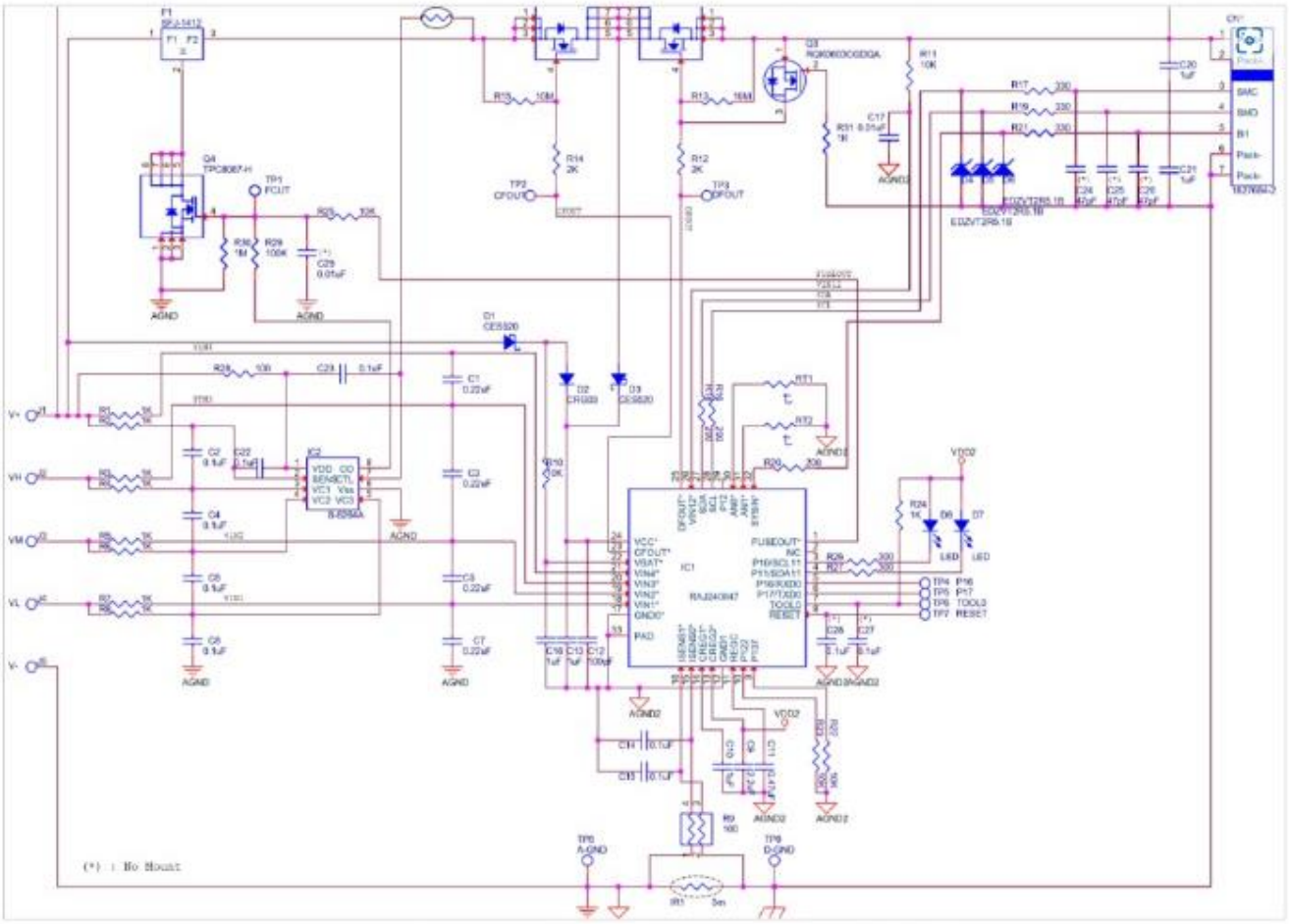
Task (1)

Name	ID
Abdullah Mohamed Ibrahim	2001826
Khaled Abdel-Monem Abdelaziz	2001605
Ziad Yasser Mohamed	2001377
Ganna Ayman Mounir	2000278
Youmna Mohamed Abdel-Fadeel	2001210
Amr Ahmed Hassan	2000957
Ahmed Mohamed Mohamed Hassan	2001920
Abdelrahman Mohamed Rafaat	2002322

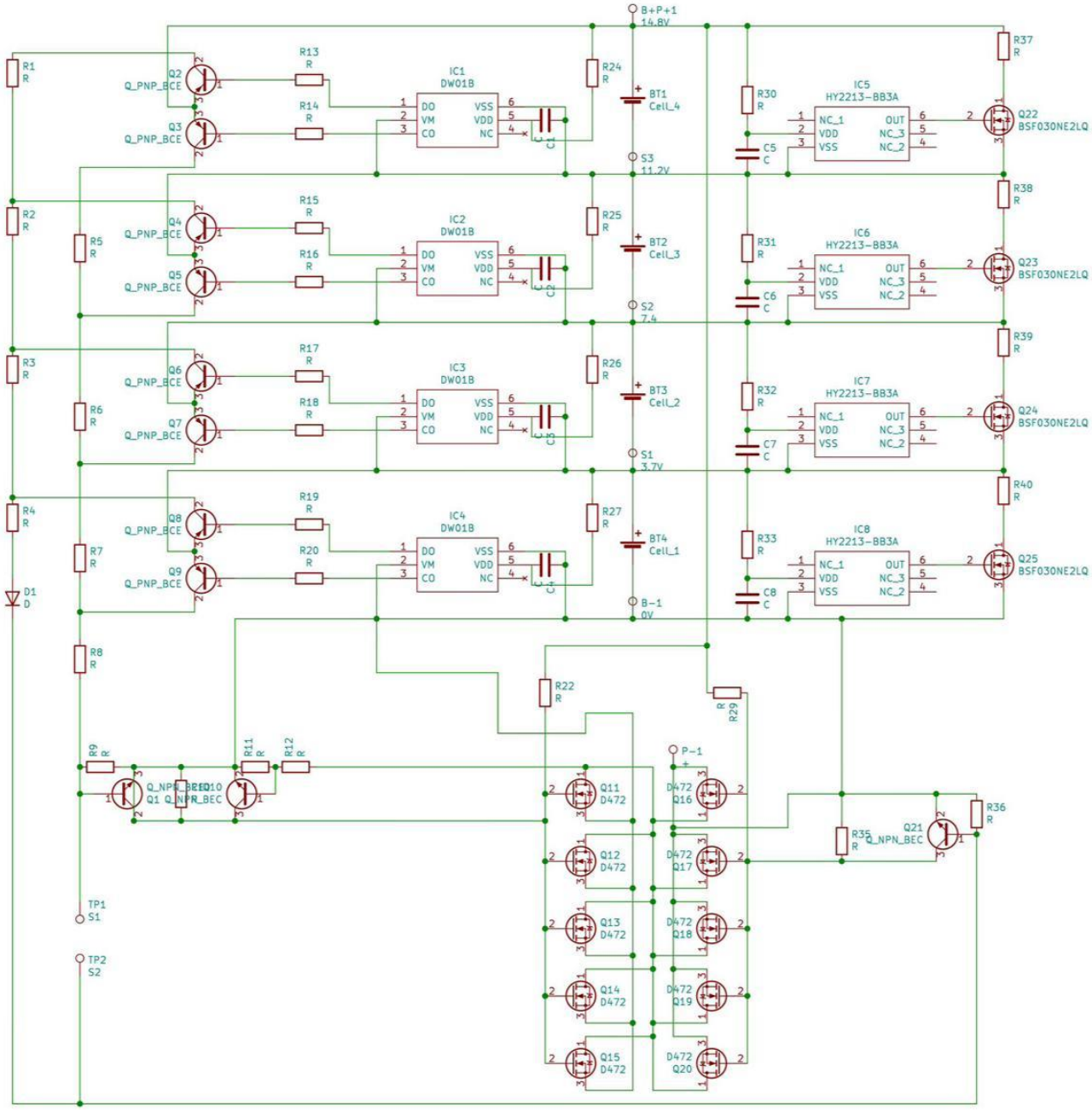
Supervised by DR. Mohamed Mostafa

Initial estimated schematic:





Detailed schematic for a 4 battery cells:

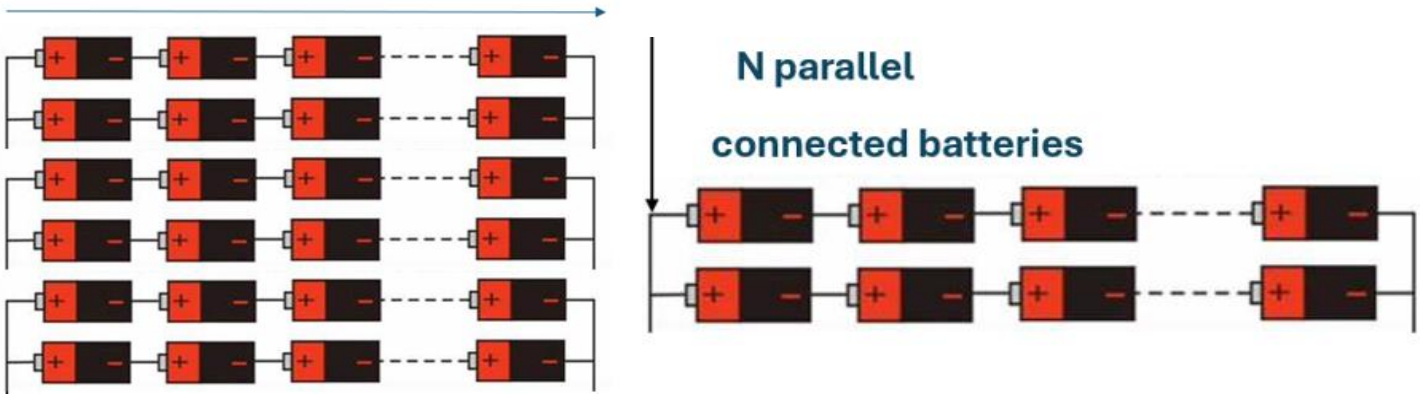


- **This schematic is of a BMS KIT that is currently on the market; our plan is to fully understand this circuit to give us the wide range to try and enhance it to reach the market's KITS efficiency**

- **We will design a battery pack providing**
  (60V max) as we plan to be placed in a Car and adjust is to its characteristics; first assumption (60V, 6.5 A) as it has a 400-watt motor

Initial Battery Pack connection:

M series connected batteries



TO reach the desired characteristics we decided that we need (MxN)

Using 3.7V 1500mAH	Using 3.7V 3000mAH
17x5 17 battery cells in series ,5 battery cells in parallel TOTAL: 85 cells Used	17x3 17 battery cells in series ,3 battery cells in parallel TOTAL: 51 cells Used

Till now the available cell in Egypt market is 3.7 V 1500mAH

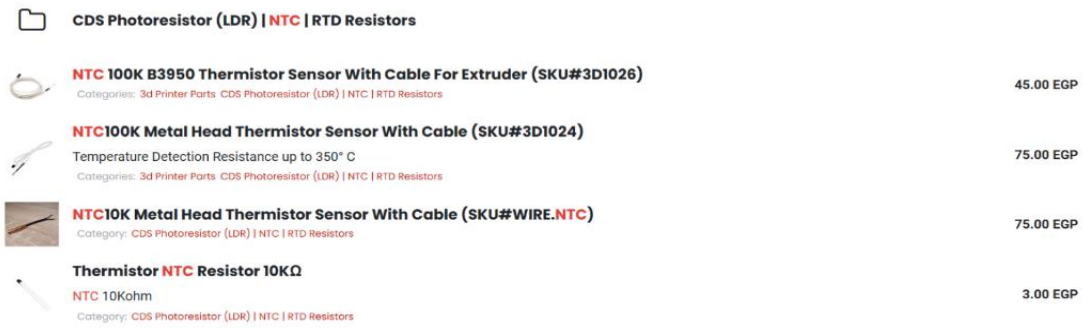
Initial Hardware estimation:

1) Temperature sensors purpose:

Monitor temperature at various points in the battery pack to prevent overheating.

a) **NTC Thermistors**

- **Connection:**
  - **Connect one terminal of the thermistor to the positive terminal of the battery cell.**
  - **Connect the other terminal to a resistor (typically around 10kΩ) which is connected to ground.**
  - **The junction between the thermistor and the resistor provides a voltage that can be read by the microcontroller's analog input.**
  - **Usage: Use the voltage divider formula to calculate the temperature based on the resistance of the thermistor.**



b) **DS18B20**

- **Connection:**
  - **Connect the VDD pin to the positive terminal of the battery cell.**
  - **Connect the GND pin to the battery cell's ground.**
  - **Connect the data pin to a digital input on the microcontroller.**
  - **Add a pull-up resistor (usually 4.7kΩ) between the data pin and the VDD.**
  - **Usage: The DS18B20 communicates over the 1-Wire protocol, allowing you to read temperature data directly.**

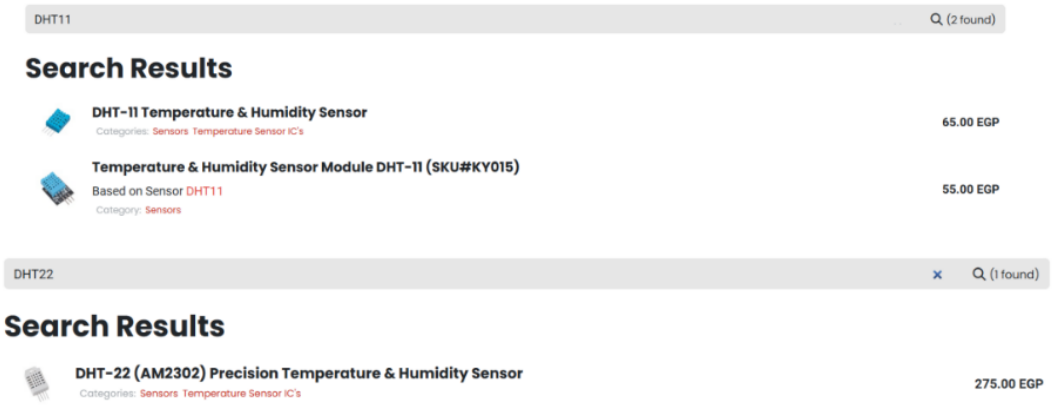
c) **DHT11/DHT22**

- **Connection:**
  - **Connect the VCC pin to the positive terminal of the battery cell.**
  - **Connect the GND pin to the battery cell's ground.**
  - **Connect the data pin to a digital input on the microcontroller.**
  - **Usage: These sensors output both temperature and humidity data in a digital format, which can be read by the microcontroller**

2)Voltage and Current Sensors

Purpose:

Measure the voltage of each cell and the total current flowing in and out of the battery.



a) **current sensor:**

The ACS712 Hall effect sensor is preferred for its isolation and safety in high current applications common in electric vehicles. Description: This sensor measures current using the Hall effect, providing an isolated output proportional to the current flowing through the conductor.

- **Connection:**

Connect the input pins of the ACS712 in series with the load (battery or motor).

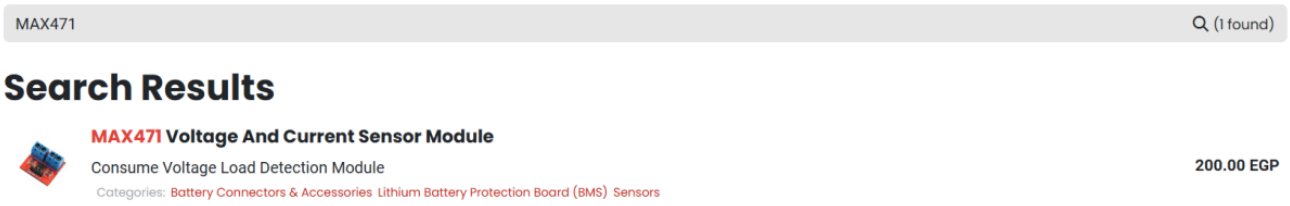
- **Connect the VCC pin to a suitable power source (e.g., 5V).**
- **Connect the GND pin to the battery ground.**
- **Connect the output pin to an analog input on your microcontroller to read the current.**

The INA219 is highly recommended for its accuracy and ease of microcontrollers with integration.

b) MAX471 Voltage Sensor

A high-side current and voltage sensor that can measure voltages up to 36V, making it suitable for battery applications with proper scaling.

- **Availability: Available in some electronics shops and online platforms.**
- **Connection:**
  - **Connect the input to the battery voltage.**
  - **Connect the output to the ADC of your microcontroller.**
  - **Make sure to scale the voltage appropriately.**



3) Microcontroller.

Generally preferred due to its higher performance and greater flexibility, making it suitable for handling complex tasks and future scalability.

Arduino is an option due to its available sources, but its low memory is a downside and lower processing speed (16 MHz).

Estimated Final Product: