To create BMS: Battery Indicator, we first got temperature using grove temperature sensor. Then we used INA 169 current sensor to measure current thru various resistors. We were tasked to create an Indicator circuit that takes current, temperature, and voltage reading from the battery and calculates the battery parameters and outputs the state of charge (SOC) and time to shut off (TTS) in real time and store it locally and online. Grove LCD was used to display output offline.

Current Sensor used was INA169, this is an Ada-fruit sensor which has 5 pins; VCC, GND, V+, V-, Vout. V+ is the positive side of the power source, and V- is the positive side of the load. These two ports connected in serial with the circuit.

In order to accomplish this task, we obtained Open Circuit Voltage(OCV) of the battery and when it was/wasn’t under load. Current sensor data connected in series with the load to measure the current being drained from the battery. The temperature sensor was connected on top of the battery to monitor the temperature when it’s being charged or discharged.

We used Arduino UNO as our microcontroller. We programmed current, temperature and voltage sensors in the analog pins. In the Arduino environment, we programmed Arduino to read the values obtained from the sensor. We decided to use this board since this board has 10-bit ADC in its analog pins, which gave us high resolution reading from the current sensor. Data obtained was were type float, and these raw data was sent to Raspberry pi thru Serial communication. Grove LCD was connected to Analog pin A4 and A5, which corresponds to SDA and SCL. These pins are used for I2C communication. I2C is a communication protocol which allows multiple slave devices to be connected with ease of two port.

Raspberry Pi is a microcomputer which operates on Raspbian OS. Raspberry Pi is capable of receiving digital data at its GPIO. Raspberry Pi model 3 has built-in WiFi and Bluetooth support, this makes raspberry pi an essential platform for IoT application. Raspberry Pi obtained Arduino data thru serial communication. SOC and OCV data were stored in raspberry pi locally as a .text file. In raspberry pi, the code was written in Python 2.7. Python is a core language of raspberry pi and has built-in capability to read serial data. Python code was able to read serial data as string format. Reading was coming at every 5-second interval. Voltage was extracted and converted to float then was compared in OCV file and its corresponding SOC was output on server and LCD. Likewise, TTS was calculated. Voltage, Current, Temperature, TTS, SOC were all put in JSON format and were uploaded to Firebase real-time database and were written to serial so it can be uploaded to LCD.

For battery we used Samsung rechargeable battery with 4.35V and LG 4.35V, we connected the current sensor and varying load to see the output current. To obtain battery characteristic we need to drain the battery at constant load and at every interval obtain the current value and integrate it. For integration, we used the trapezoidal method, for this method we compared it with the past value and assumed it create trapezoid and calculated the area, this returned roughly the integration of current. From this, we obtained the capacity of the battery, and from the capacity, we were able to obtain charge left in the battery. From these data, we were able to map OCV and SOC of the battery. This is called the coulomb counting method, we used this method to map out 3.7V Li-ion battery, for 4.35V battery SOC-OCV data was given by Dr.Bala.