BMS Error process

Data Structure used

BMS\_ERROR array containing number of elements equals to error types

Error Type definitions

ERROR State definitions

Process:

1. Process BMS error frame and update error type states. Done
2. Save error in eeprom

**BMS Error can occurs in following ways:**

1. Error occurred and cleared
2. Error occurred and remained
3. Error occurring and clearing multiple times

**EEPROM write method**

Case 1 eeprom can be written when error occurred

Case 2 eeprom should be written only when error occurred first time

Case 3 eeprom should be written when error occurred first time

Possible method: compare current with last saved method if different then only insert it in queue.

**Counter for error count**

**Only save for current ignition cycle**

When ignition switches off then clear last data and write new data.

**QUEUE implementation**

Latest value is pushed in LSB

insert

((Bms\_error64 << 8) | error8)

Read

(Bms\_error64 & 0xFF)

**QUEUE implementation with Counter**

Issue when multiple error in same frame

If counter is used then this problem will be solved

**Method to insert error code with counting:**

Check if error code already exist in queue

If exist then only increment the value associated with it

If not exist then push new error with count value of 1.

**When to update eeprom value**

Write to EEPROM - Ignition off

Read from EEPROM - ignition ON

Issue: if we read from eeprom when ignition off then we get old ignition on data

Sold: when