# Can library system requirements:

1. Get data from can network and parse and store it.
2. Maximum delay for getting the can parameters should be no more than 1 second
3. Parameters required from BSS hardware:
   1. Total charging slots – not needed as later decided
   2. Status of each slot ID  (unoccupied/ occupied with battery’s ID):
      1. Battery’s unique ID - done
      2. BSS’ unique ID (for authorization purpose) – not available
      3. Slot ID – done
      4. State of Charge(SOC) - done
      5. State of Health(SOH) - done
      6. Battery’s Temperature  - done
      7. Battery’s Voltage - done
      8. Battery’s Current - done
      9. Rate of charge – not needed as later decided
      10. Battery’s age – not needed as later decided
      11. Battery’s type – not needed as later decided
      12. Battery’s max cycles – not needed as later decided
      13. Battery’s utilized cycles - done
      14. String cell level data – not needed as later decided
   3. Total input current – not available
   4. Input Voltage – not available
   5. Input power – not available
   6. Power factor of BSS – not available
4. Parameters required from the EV:
   1. Vehicle’s unique ID (for authorization purpose) – not available
   2. Motor’s voltage - done
   3. Motor’s current - done
   4. Motor Control Unit’s temperature - done
   5. Motor Control Unit’s RPM (rev per minute) - done
   6. Battery Status(for each battery):
      1. Battery’s unique ID - done
      2. Vehicle’s unique ID (for authorization purpose) – not available
      3. State of Charge(SOC) - done
      4. State of Health(SOH) - done
      5. Battery’s Temperature  - done
      6. Battery’s Voltage - done
      7. Battery’s Current - done
      8. Rate of charge – not needed as later decided
      9. Battery’s age – not needed as later decided
      10. Battery’s type – not needed as later decided
      11. Battery’s max cycles – not needed as later decided
      12. Battery’s utilized cycles - done
      13. String cell level data – not needed as later decided

# Implementation of library:

1. Initializing of CAN protocol includes the setting of speed and initialization of mcp2515. The first action performed is the checking of battery ids. Right now the code for checking battery ids in ev is tailored to the bss. The max count of can requests needs to change. Since this is the only CAN REQ event, a timeout for response is set.
2. The send\_msg was developed for testing purposes. It is outdated and needs to be updated in case it is used.
3. The receive msg function is a generic function that is able to read all types of can data be it response of a request or a periodic can message. It further branches into several functions based on the id of received message. The id of received message declares what the data represents and how it is to be interpreted. When valid data is received the function reads the data and updates the relevant data structure variables.
4. The two main data structures of CAN library are BmsData and EvData. Their respective parameters can be found in library’s header file.
5. The parsing of can data is based on the format defined by the Koreans. Parsing includes manipulation of available 8 bytes of data in a single can message.