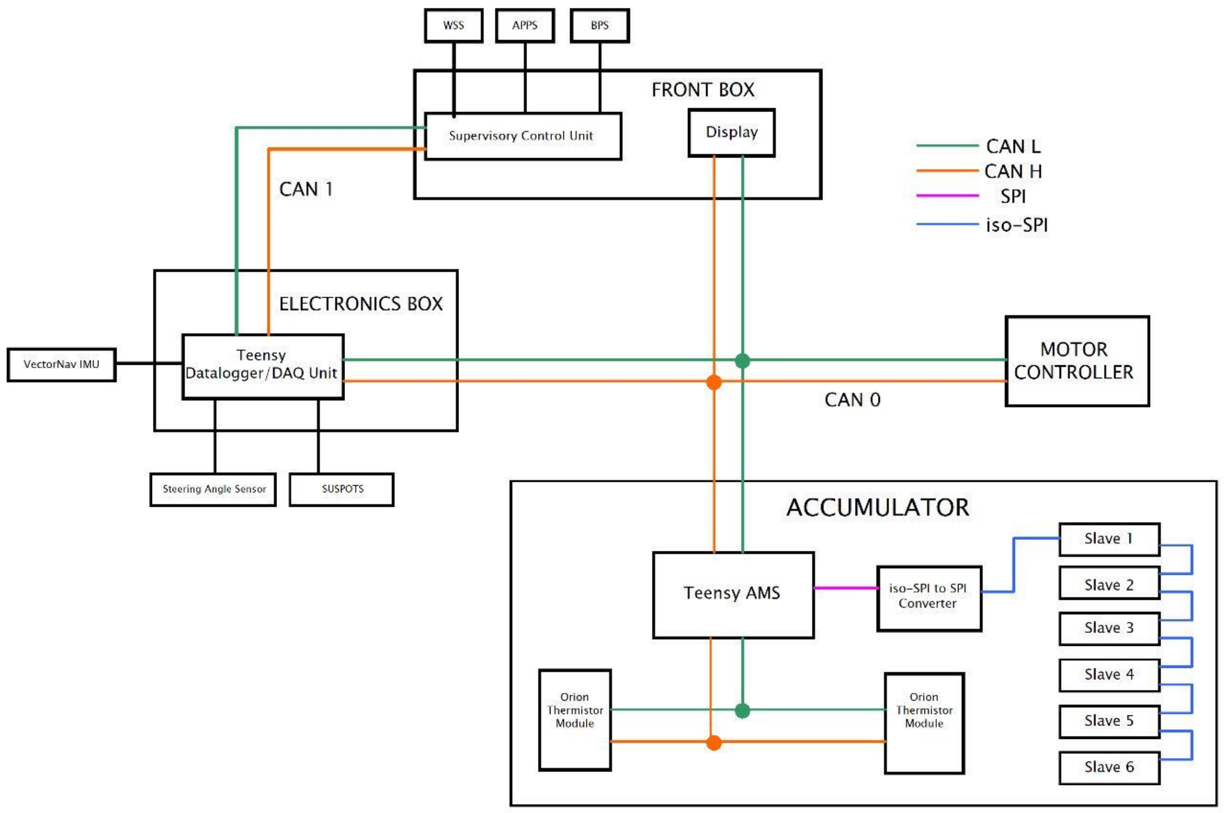
Control Systems Overall Section Concept

Year	2023-2024
Section	Controls
Member(s) Responsible	Abhay, Sanskar, Yash

System Integration



Concept

The Control Systems design of FMe23 involves the design of Accumulator Management System (AMS), Supervisory Control Unit (SCU), Data Acquisition System, Motor Controller and a Display System all integrated together on the car.

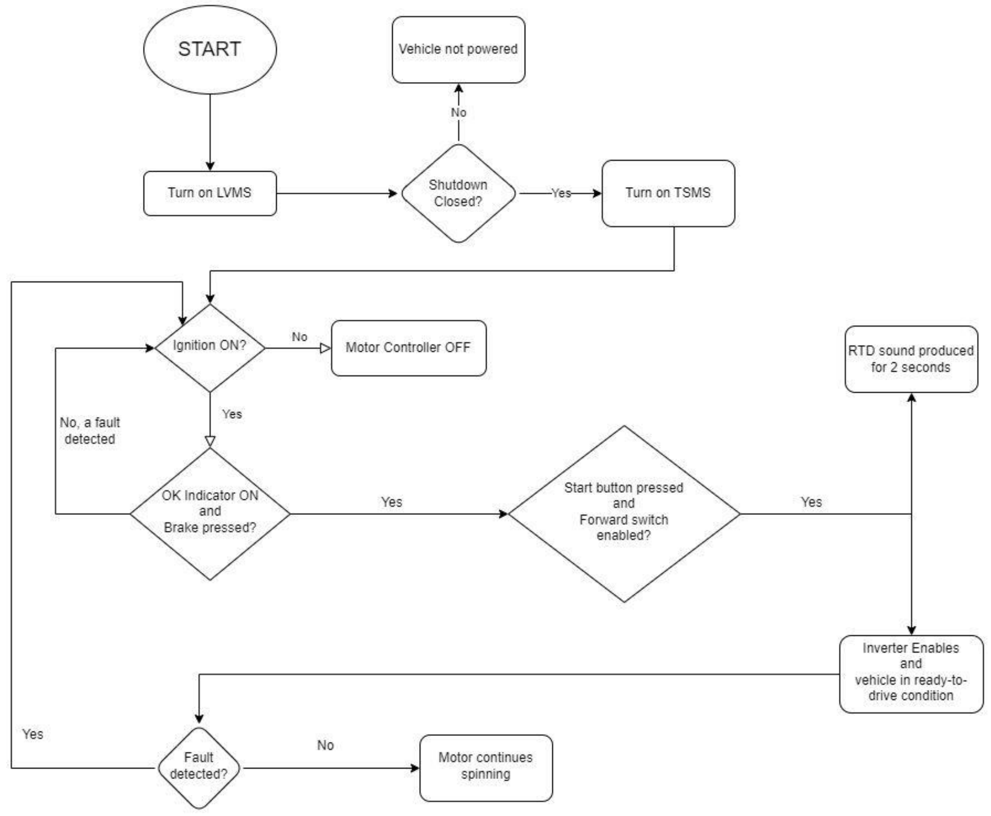
Supervisory Control Unit

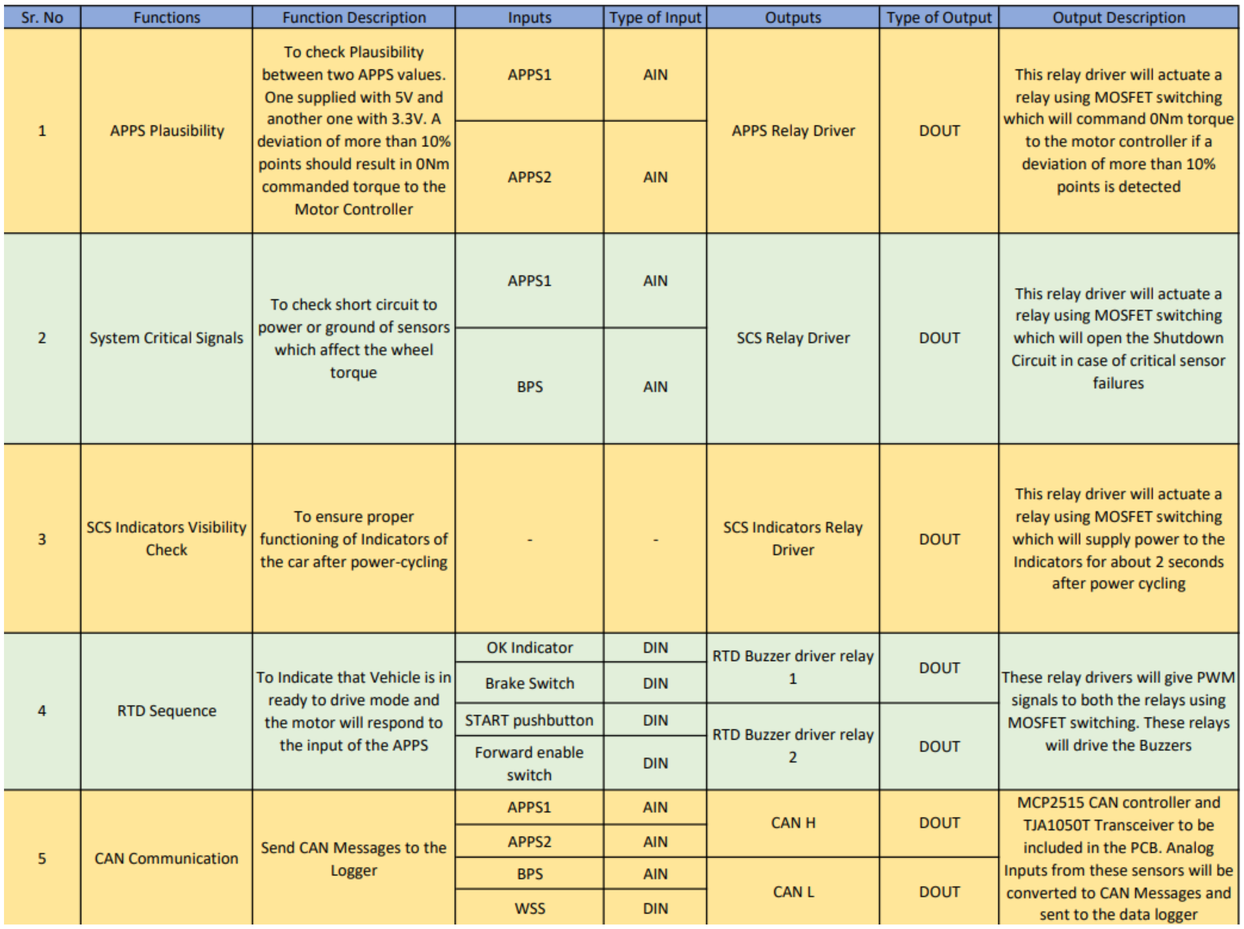
The main job of the Supervisory Control unit is to perform Implausibility Checks on the two APPS, check for brake switch input, perform the startup sequence for the Ready to Drive sound and to sense all the 5v Sensors present on the car like APPS, BPS and the Wheel Speed Sensors.

Arduino Mega was chosen as the microcontroller for SCU among boards like Teensy 4.1, MINI M4 STM32, STM32 Nucleo,

Reason for this design decision was most of the boards mentioned above are not 5V tolerant making it difficult to sense 5v Wheel speed sensors and sense 5v digital inputs and outputs while STM boards that are 5v tolerant will take a longer time to procure.

Below is the Flow Chart of the working of the SCU





Sr. No	Functions	Function Description	Inputs	Type of Input	Outputs	Type of Output	Output Description
1	APPS Plausibility	To check Plausibility between two APPS values. One supplied with 5V and another one with 3.3V. A deviation of more than 10% points should result in 0Nm commanded torque to the Motor Controller	APPS1	AIN	APPS Relay Driver	DOUT	This relay driver will actuate a relay using MOSFET switching which will command 0Nm torque to the motor controller if a deviation of more than 10% points is detected
			APPS2	AIN			
2	System Critical Signals	To check short circuit to power or ground of sensors which affect the wheel torque	APPS1	AIN	SCS Relay Driver	DOUT	This relay driver will actuate a relay using MOSFET switching which will open the Shutdown Circuit in case of critical sensor failures
			BPS	AIN			
3	SCS Indicators Visibility Check	To ensure proper functioning of Indicators of the car after power-cycling	-	-	SCS Indicators Relay Driver	DOUT	This relay driver will actuate a relay using MOSFET switching which will supply power to the Indicators for about 2 seconds after power cycling
4	RTD Sequence	To Indicate that Vehicle is in ready to drive mode and the motor will respond to the input of the APPS	OK Indicator	DIN	RTD Buzzer driver relay 1	DOUT	These relay drivers will give PWM signals to both the relays using MOSFET switching. These relays will drive the Buzzers
			Brake Switch	DIN			
			START pushbutton	DIN	RTD Buzzer driver relay 2	DOUT	
			Forward enable switch	DIN			
5	CAN Communication	Send CAN Messages to the Logger	APPS1	AIN	CAN H	DOUT	MCP2515 CAN controller and TJA1050T Transceiver to be included in the PCB. Analog Inputs from these sensors will be converted to CAN Messages and sent to the data logger
			APPS2	AIN			
			BPS	AIN	CAN L	DOUT	
			WSS	DIN			

Accumulator Management System (AMS)

AMS is responsible for monitoring, controlling, and optimizing the performance, safety, and longevity of the Accumulator.

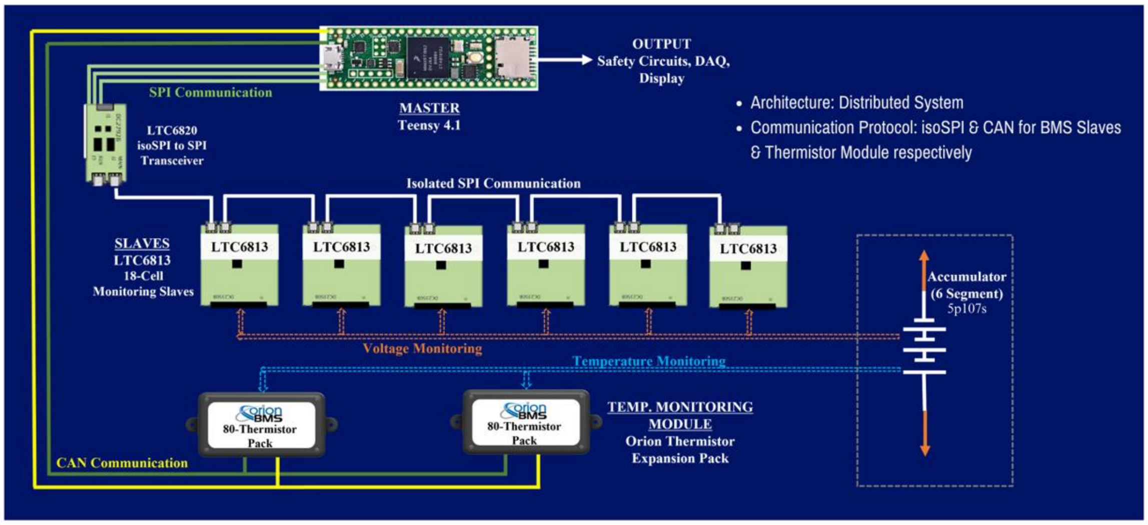
Tasks of AMS

1. Establish SPI Communication with the IsoSPI-SPI Transciever and monitor the cell voltage from 107 cell taps from 6 LTC6813 slaves.
2. Establish CAN Communication with the Thermistor Modules and monitor 160 thermistors over CAN
3. Perform a basic SoC estimation based on the Voltage method by using the cell voltages and their discharge curve. Estimation methods like coulomb counting cant be used since the LTC demoboards dont have any provision to measure cell currents.
4. Log the cell voltages and the temperatures using the onboard SD card. This logging is seperate from the Logging in the DAQ system making it completely independent.
5. Send a periodic CAN message outside the Accumulator using the 2nd CAN controller of Teensy. This CAN message will have Highest, Lowest, Average Cell voltages and Cell temps of the pack for use in displaying on the Display.
6. Motor Controller Derating is performed via a CAN message to the PM100DZ as it senses the pack current using a current sensor.

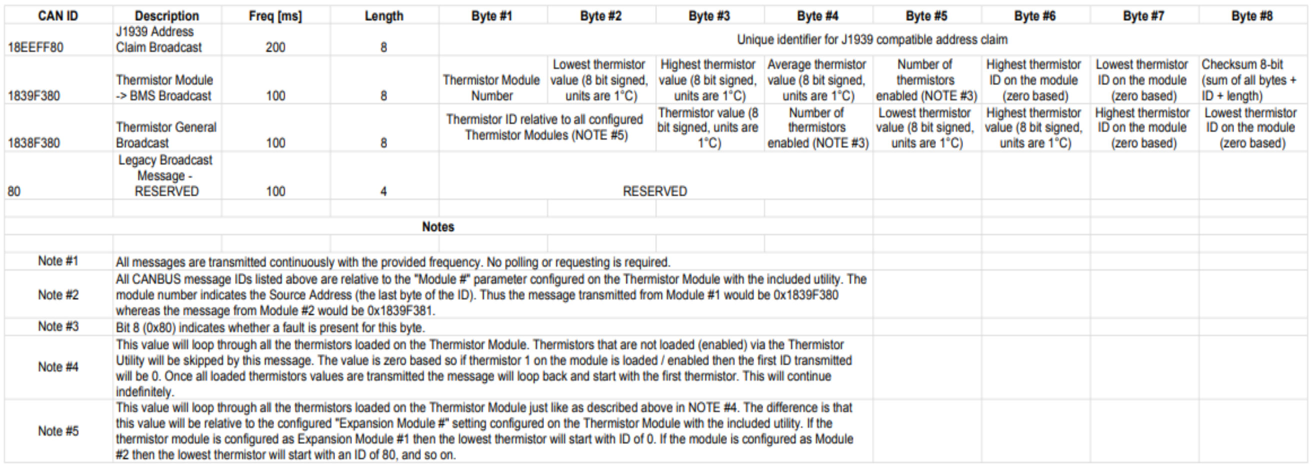
7. Send an output to the AMS Latch PCB when Cell Voltages or Temps exceed their limits.

To Perform all these tasks, we chose to use Teensy 4.1 as the AMS master due to its smaller form factor, High CPU speed, 3 CAN controllers and onboard SD Card all at a cheaper cost. The above tasks will be performed over FREERTOS.

System Overview



Thermistor Modules CAN Message order



CAN ID	Description	Freq [ms]	Length	Byte #1	Byte #2	Byte #3	Byte #4	Byte #5	Byte #6	Byte #7	Byte #8
18EEFF80	J1939 Address Claim Broadcast	200	8	Unique identifier for J1939 compatible address claim							
1839F380	Thermistor Module -> BMS Broadcast	100	8	Thermistor Module Number	Lowest thermistor value (8 bit signed, units are 1°C)	Highest thermistor value (8 bit signed, units are 1°C)	Average thermistor value (8 bit signed, units are 1°C)	Number of thermistors enabled (NOTE #3)	Highest thermistor ID on the module (zero based)	Lowest thermistor ID on the module (zero based)	Checksum 8-bit (sum of all bytes + ID + length)
1838F380	Thermistor General Broadcast	100	8	Thermistor ID relative to all configured Thermistor Modules (NOTE #5)		Thermistor value (8 bit signed, units are 1°C)	Number of thermistors enabled (NOTE #3)	Lowest thermistor value (8 bit signed, units are 1°C)	Highest thermistor value (8 bit signed, units are 1°C)	Highest thermistor ID on the module (zero based)	Lowest thermistor ID on the module (zero based)
80	Legacy Broadcast Message - RESERVED	100	4	RESERVED							
Notes											
Note #1	All messages are transmitted continuously with the provided frequency. No polling or requesting is required.										
Note #2	All CANBUS message IDs listed above are relative to the "Module #" parameter configured on the Thermistor Module with the included utility. The module number indicates the Source Address (the last byte of the ID). Thus the message transmitted from Module #1 would be 0x1839F380 whereas the message from Module #2 would be 0x1839F381.										
Note #3	Bit 8 (0x80) indicates whether a fault is present for this byte.										
Note #4	This value will loop through all the thermistors loaded on the Thermistor Module. Thermistors that are not loaded (enabled) via the Thermistor Utility will be skipped by this message. The value is zero based so if thermistor 1 on the module is loaded / enabled then the first ID transmitted will be 0. Once all loaded thermistors values are transmitted the message will loop back and start with the first thermistor. This will continue indefinitely.										
Note #5	This value will loop through all the thermistors loaded on the Thermistor Module just like as described above in NOTE #4. The difference is that this value will be relative to the configured "Expansion Module #" setting configured on the Thermistor Module with the included utility. If the thermistor module is configured as Expansion Module #1 then the lowest thermistor will start with ID of 0. If the module is configured as Module #2 then the lowest thermistor will start with an ID of 80, and so on.										

For Thermistor CAN communication to the AMS, the module sends out a continous CAN message with ID-1838F380 whose first byte iterates over the thermistors from 0-80 which will be read by the AMS in a for loop and assigned to an array of size 80 which when fills up will be displayed on the Serial Monitor while also being logged. Since these two task of reading CAN and logging will use the critical section of the code, Mutexes or Semaphores will be used.

The same step above will be executed for the Module #2

Data Acquisition System(DAQ)

DAQ is responsible for sensor data acquisition, CAN communication with the Motor Controller (Diagnostic Data), CAN communication with the SCU

Tasks of DAQ

1. Acquire data from 4 Linear Suspension Potentiometers and 1 Steering angle sensor at a sampling rate of 100Hz.
2. Establish CAN1 communication with the Motor Controller and the AMS and recieve the Diagnostic Data.
3. Establish CAN2 communication with the SCU and recieve the APPS, BPS, Ready to Drive State and the wheel speed sensor data.
4. Log Data from all the above tasks into multiple files on the onboard SD Card
5. To Run the Motor in CAN Control Mode, the board will send periodic CAN messages having the torque command and inverter enable data while also monitoring the state of the inverter. (In Development and Optional)

The torque command will be decided by making a APPS Map that works in a similar way to that of the way the Motor Controller works in the VSM State.

Teensy 4.1 is chosen as the board for Data Acquisition system

Motor Controller

VSM Mode

The default method of controlling the motor in car will be using its VSM State mode with the Inverter Enable Sequence being

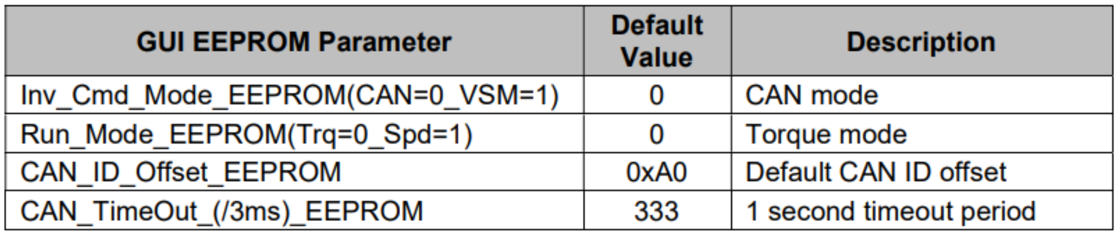
1. Shutdown Circuit Closed
2. Precharge Sequence completed and the MC being in the VSM state-6
3. Brake input is grounded via the brake switch
4. FWD enable which is controlled by the shutdown circuit while also working as a switch place on the dash
5. Start Pulse from the Start Pushbutton

When this sequence is completed the Car and the motor are both in the ready to drive state. The torque is controlled via the signal from the APPS directly connected to the MC.

CAN Mode

When the MC is going to be used in CAN Mode the above sequence for VSM is no longer viable.

Before the CAN mode is used these 4 parameters need to be changed using the software manually.



GUI EEPROM Parameter	Default Value	Description
Inv_Cmd_Mode_EEPROM(CAN=0_VSM=1)	0	CAN mode
Run_Mode_EEPROM(Trq=0_Spd=1)	0	Torque mode
CAN_ID_Offset_EEPROM	0xA0	Default CAN ID offset
CAN_TimeOut_(/3ms)_EEPROM	333	1 second timeout period

Startup Sequence

1. Shutdown Circuit is closed with the Motor controllller powered on
2. Ensure the CAN communication with the MC is established
3. Precharge sequence completed.
4. Switch on the Inverter Enable Safety Switch on the dashboard
5. Ensure the Car is in Ready to drive state
6. Send out inverter Disable command
7. Send inverter Enable command that is present in the CAN message 0xC0.
8. Send Continuous CAN torque command from the DAQ board within the timeout period.

To prevent Data corruption on the CAN bus line currently we have only thought of CRC checking.

To run CAN mode only one message is to be sent to the MC to enable and run the motor throughout.

The other messages can be used to change the EEPROM parameters of the Motor Controller on the run but we have decided not to implement them.

The Command message is as follows

2.2 Command Message

The Command Message is used to transmit data to the controller. This message is sent from a user-supplied external controller to the motor controller. The Control Message (0x0C0) is used to operate the controller via the CAN interface.

0x0C0 – Command Message (Data Length = 8 bytes)

Byte.Bit	Name	Format	Description
0,1	Torque Command	Torque	Torque command used when in torque mode. When in Speed Mode the Torque Command values become a feedforward to the Speed Regulator (see Using Speed Mode manual).
2,3	Speed Command	Angular Velocity	Speed command used when in speed mode. Starting in version 2048 and 651D any Speed Command transmitted while in Torque mode will over-ride the Max Speed EEPROM parameter and provide a new maximum speed limit. The maximum speed limit will revert to the default EEPROM parameter value when the inverter returns to Torque mode.
4	Direction Command	Boolean	0 = "Reverse" 1 = "Forward" See section 2.3.2.2 for further definition of direction.
5.0	Inverter Enable	Boolean	0 = Inverter Off, 1 = Inverter On
5.1	Inverter Discharge[8]	Boolean	0 = Disable Discharge, 1 = Enable Discharge
5.2	Speed Mode Enable	Boolean	0 = Do not over-ride mode 1 = If controller is in torque mode then controller will change to speed mode. This is a mode over-ride bit that will change the mode from torque to speed only. It does not change the mode from speed to torque. See manual Using Speed Mode for more information.
6,7	Commanded Torque Limit	Torque	If set to 0, the default torque limits sets in the EEPROM parameters are used. If set to a positive number then the Motor and Regen Torque limits are set to the torque value sent.

Concept Review: