A. Overview

The objective is to control the contactors that connect the battery string to the motor inverter (DMOC). A key part of this is pre-charging the DMOC capacitor so that when the contactor is closed the current surge is small.

Also involved is to detect special conditions and failures. E.g. if the battery string voltage is too low a command to connect will put the ContactorTask state in a fault condition and not respond to connect commands.

This task organization is based on using STM32CubeMX and FreeRTOS, and be event driven.

B. Hardware

1. Several contactor configurations are accommodated--

a. Two contactors

1). With, or without, auxiliary contacts.

2). With, or without, pwm

Contactor #1 connects the battery string plus to DMOC plus.

Contactor #2 connects battery string minus to DMOC minus.

Pre-charge resistor is across contactor #2 contacts.

b. One contactor, one small pre-charge relay

1). Contactor with, or without, aux contacts and with or without pwm.

The contactor connects the battery string plus to DMOC plus.

2) Small relay connects battery string plus to pre-charge resistor and the pre-charge resistor connects to the DMOC plus.

The DMOC minus connects directly to the battery string minus.

For program purposes, the contactor is designated as contactor #2 (since it closes the high current path to the DMOC after pre-charge), and the relay designated as contactor #1 since closing it begins the pre-charge.

2. Isolated high voltage

High voltage is measured and sent to the ContactorTask via an isolated uart RX. Three voltages are from a common ground on the battery string are measured and sent as a line (maybe binary).

a. “hv1” - battery string plus

b. “hv2” - DMOC plus

c. “hv3 ” - DMOC minus

hv3 measures the voltage across the pre-charge resistor when there are two contactors, i.e. the voltage across contactor #2 contacts.

The difference between hv1 and hv2 measures the voltage across the pre-charge resistor in the one contactor configuration.

3. DMOC enable

A small fet pulls DMOC enable line, or possibly a small relay for daisy-chaining a DMOC enable.

4. CAN

A non-isolated CAN driver.

5. Current sensors

Provision for two Hall-effect current sensors.

a. Battery string current

b. spare

C. ContactorTask specifics

Keep-alive (KA)--

The keep-alive CAN msg sent to the ContactorTask by the function controlling the contactor carries the command as well as periodically confirms that there is control.

At initialization, a keep-alive/command rtos software timer is started and is periodic. Every incoming keep-alive/command CAN msg carries the current command, e.g. "connect" (bit ON)/"disconnect" (bit OFF), and "reset/fault", e.g. (bit ON to command a reset and remove from locked up fault state), and (bit OFF for not reset).

Every incoming KA msg results in a CAN msg response that conveys the current status info. ContactorTask.h , comment section starting (line 19) shows the payload detail of the response (as well as the other CAN msgs).

The periodic timer is reset every time a KA CAN msg arrives. If it times out, meaning control has been lost, it will transition the state to "FAULTING" which then goes to "FAULTED" after a delay for contactors to open.

The Keep-alive timeout is independent of the current state, except "FAULT" (since there is no point to repeating the delay for contactors to open in "FAULTING").

There is an issue here about whether the KA timeout should go to "DISCONNECTING" and onward to "DISCONNECTED," rather the FAULT. To exit the FAULT state requires a RESET command; the DISCONNECTED state exits when a CONNECT command is received. So, if a KA timeout occurs and it ends up in FAULT, then to get started requires a RESET command plus a CONNECT command. If it only goes to DISCONNECT, then anytime the KA msgs resume a connect command can be sent.

By having the KA timer be periodic, status msgs will be sent every timeout, thus making it a heartbeat with status msgs when the controller node issuing the KA msgs, e.g. MMC, is not sending or something else is not right (mainly a problem during development).

KA status msgs are sent immediately when there is a change in status.

If the MMC forwards status msgs the MC, or maybe PC, can display status for critical errors, as well as take action.

Items--

On the sketch BEGIN, i.e. it starts in the initialized state--

We start with contactors open, etc. and the state being "disconnected."

**DISCONNECTED state**

This state is changed when:

1) Battery string voltage is less than require to start a launch--

Set code number for battery string low.

New state: FAULTING

2) If Aux contacts are present on contactor #1, AND Aux 1 does not show OFF--

Set timer2 for longest the two contactors opening delay

New state: FAULTING with code number for this fault.

3) If Aux contacts are present on contactor #2, AND Aux 2 does not show OFF. (Obviously, when there is one contactor and a pre-charge relay, there would be no Aux 2 contacts)--

Set timer 2 for contactor #2 opening delay

New state: FAULTING with code number for this fault.

4) CAN command (keep-alive) msg "connect" is received.

Set software timer2 for contactor #1 closure delay--

(approx 25 ms and if it has aux contacts this parameter might have to include

some delay if the aux contact close a bit behind the main contacts)

Set connecting substate to C1

New state: CONNECTING

**CONNECTING state**

**substate C1** closing contactor #1

This state changes when:

1) Timer2 times out

Contactor #1 should now be closed, contactor #2 still open.

If aux1 present, and is not ON, then

Set code for aux1 not closing

New state: FAULTING

else

If two contactor config, and hv2 not approx equal hv1, then

Set code for contactor #1 didn’t close

New state: FAULTING

Set timer2 to minimum pre-charge duration (approx 3 secs)

New state: substate C2

**substate C2** charging DMOC capacitor

This state changes when:

1) Timer2 times out.

Set timer for limit to extended pre-charge duration (4 secs?)

New state: substate C3

**substate C3** charge DMOC cap until cutoff voltage

This changes state when:

1) If two contactors

hv3 is less than cutoff voltage (e.g. 10v)

else

(hv1-hv2) less than cutoff voltage

Set timer2 for contactor #2 closing delay

New state: substate C4

**substate C4** contactor #2 closing

This changes state when:

1) If aux2 present and aux2 not closed

Set fault code: contactor #2 aux not closing

New state: FAULTING

2) Timer 2 times out.

If one contactor config,

de-energize contactor #2 (pre-charge relay)

New state: CONNECTED

**CONNECTED state**

This changes state when:

1) KA disconnect or reset commands received

Set timer2 to longest of contactors open delays.

New state: DISCONNECTING

**DISCONNECTING state**

Changes state when:

1) timer2 times out.

New state: DISCONNECTED

**FAULTING state**

Changes state when:

1) timer2 times out

New state: FAULTED

**FAULTED state**

Changes state when:

1) KA command reset received

Clear fault code

New state: DISCONNECTED