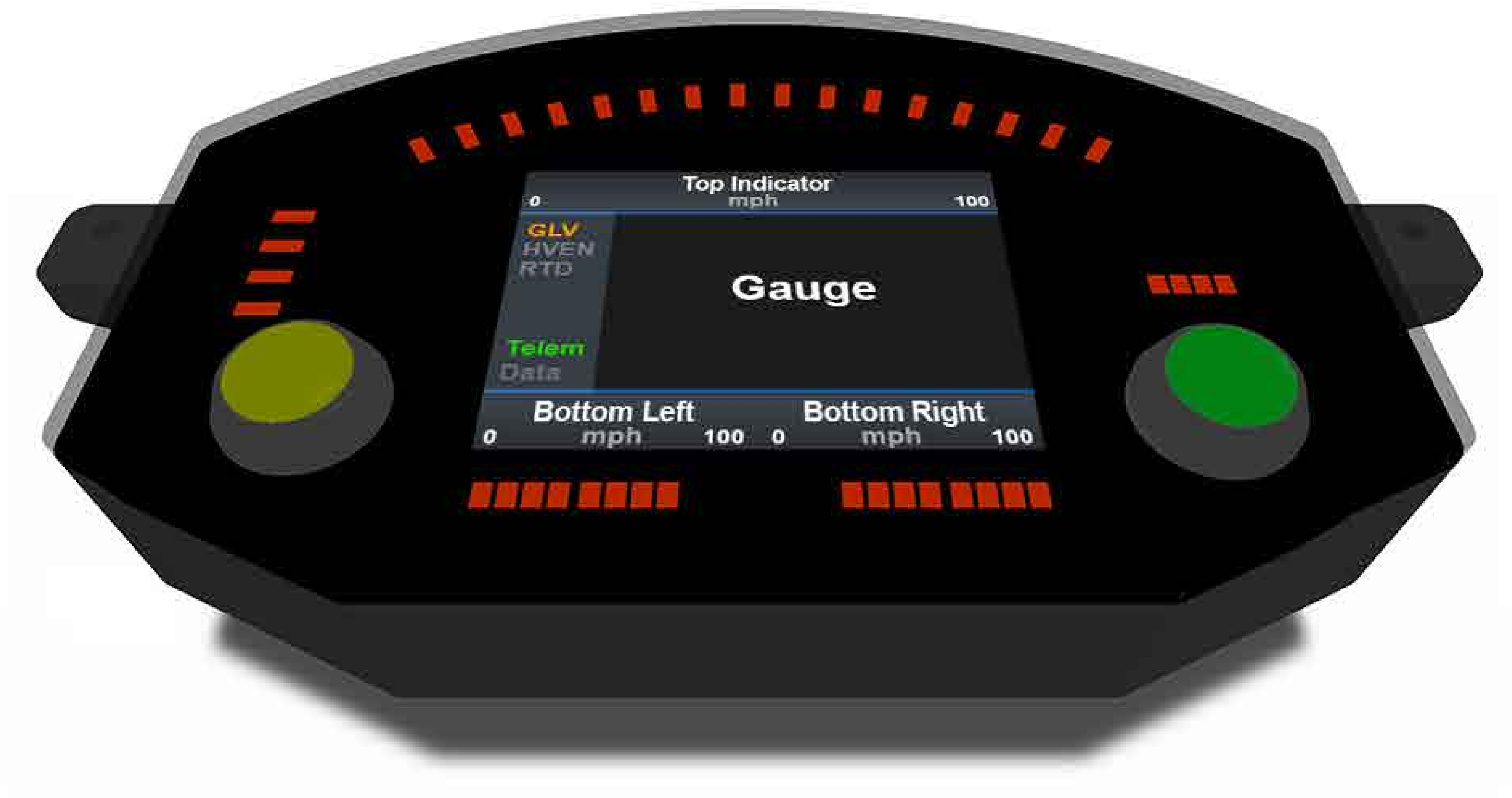
Driver Interface Module for an Electric Racecar

18-549 Embedded Design Capstone, Spring 2016

Project and Software Overview

Introduction

The Driver Interface Module is an ARM Cortex M4-powered dashboard for a Formula SAE electric racecar. It is designed to communicate on a CAN network, and features an LCD display, two configurable RGB LED bargraphs, and a configurable RGB LED arc display for system feedback. It is specifically engineered for CMR 16e -- Carnegie Mellon Racing's 2016 electric racecar -- but is adaptable for use in any CAN-connected system.



The global vehicle state is coordinated by the Safety Module, which serves to monitor the safety state of the car and trigger state changes. The Driver Interface Module requests state transitions from the Safety Module based on user input, and notifies the user of errors. As the system must be robust and shut off only when either the vehicle or the driver are in danger, a variety of errors, both critical and non-critical, can cause the system to shut down. Four LEDs on the left side of the dashboard indicate error type, and a single four-LED bargraph on the right of the dashboard blinks to signal the driver to egress in case of an emergency error.

Issues can be quickly diagnosed with verbose error messages displayed on the dashboard, minimizing repair time in case of a fault.

Task Name	Priority	Frequency [Hz]	Duration [µs]	Utilization [%]
CAN IRQ	1	8000	3	2.4
CAN Hearbeat	2	100	20	0.2
Button Polling	3	100	5	0.05
LCD Update	4	60	5000	30
LED Update	5	6	30000	18
			Total Util:	50.65

Team Members

Shepard Emerson
semerson@andrew.cmu.edu

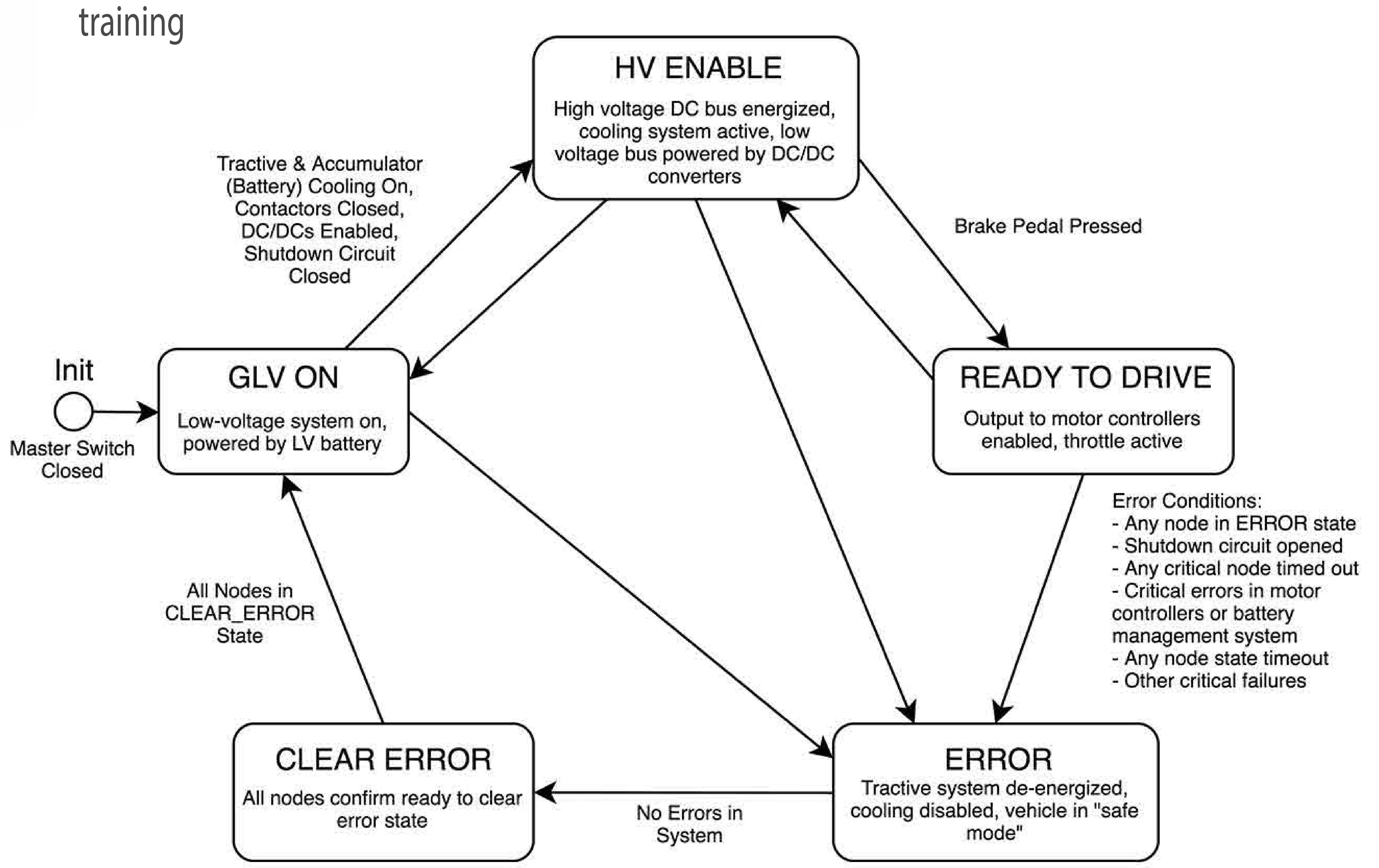
Tom Eliot
tke@andrew.cmu.edu

Daniel Gorziglia
dgorzigl@andrew.cmu.edu

Daniel Haddox
dhaddox@andrew.cmu.edu

Network Architecture

The Driver Interface Module is one node in a distributed system comprised of ten CAN-connected controllers -- a battery management system, two motor controllers, and seven custom designed sensing, telemetry, and cooling nodes. All custom nodes run FreeRTOS, and time-triggered communication is used to coordinate system state changes. The Driver Interface Module provides the user with control over system state and parameters (such as traction control coefficients and max speed) and enables easy real-time driver feedback for more efficient driver training



HV ENABLE
High voltage DC bus energized, cooling system active, low voltage bus powered by DC/DC converters
Tractive & Accumulator (Battery) Cooling On, Contactors Closed, DC/DCs Enabled, Shutdown Circuit Closed
Brake Pedal Pressed
Init
Master Switch Closed
GLV ON
Low-voltage system on, powered by LV battery
READY TO DRIVE
Output to motor controllers enabled, throttle active
Error Conditions:
- Any node in ERROR state
- Shutdown circuit opened
- Any critical node timed out
- Critical errors in motor controllers or battery management system
- Any node state timeout
- Other critical failures
All Nodes in CLEAR_ERROR State
CLEAR ERROR
All nodes confirm ready to clear error state
No Errors in System
ERROR
Tractive system de-energized, cooling disabled, vehicle in "safe mode"

Task Scheduling

Because the Driver Interface Module is a safety critical node, care must be taken to ensure that all tasks are be schedulable and meet deadlines. RMS scheduling with preemption is implemented using FreeRTOS. Five tasks are used to perform all of the module's functions -- a CAN interrupt request handler, a CAN heartbeat task, a button polling task, an LCD update task, and an LED update task. The priorities, frequencies, and durations of the tasks are listed in the table to the left, as well as the total utilization of real-time tasks. Because the total utilization is less than 69.3%, schedulability is guaranteed

Carnegie Mellon
Electrical & Computer ENGINEERING
Carnegie Mellon Racing
altia