FRESTA - future TODOs

**CAN:**

* 125 kbps CAN transmission doesn’t seem to work (20, 25, 250, and 500 work fine). Is there some problem on the Linux side?
* Can\_SetControllerMode() is currently a shell function, only changing the state names, but not really changing the functionality in each state
* Several functions, declared in Can.h (for example Can\_MainFunction\_Write()), are not implemented.
* Can\_FindHoh() is not used in our implementation. Perhaps we should use HOHs (or HTH) for storing data (instead of the CanMessage-structure) and communicating it to the CanIf-layer.
* Can\_Write() is not using its first parameter (HTH) in our implementation

**Performance:**

* Steering delay – why?
* Make Leds work
* Add steer select logic
* Add ports in PIRTE-SWC for receiving speed&steer signals
* Linux node only connects to WiFi-spot at start-up (should search for a connection continuously)

**Cleaning:**

* Remove extern\_defs.h
* Remove #if VCU (only used in java.c)
* Move java.c from examples to a more central place
* What to do with vcu/scu\_platform.c?
* Review CanTp
* Review IMU
* Review Actuator\_Led
* Check that all HW is working
* Move ADC to another chip select, CE1
* Update the xls-file on Moped-wiki with CE1 and CAN-interrupt pin
* Clean up apps. Do they all work? Can they be compiled using javac?
* Can zip-creator be omitted (squawk might need adjustment)?
* Ecm-linux does not compile (Main.java calls IoTManager(publisher). This constructor is not implemented in the last IoTManager-version. Why did we switch to IoTManager(publisher, receiver)?) Ecm.java is also affected (iotManager.setEcm(this) for some reason??). Also CarFactory.java.
* Clean up MessageType.java

**Refactoring:**

* Move out PIRTE-code from squawk
* Find a place for api/ (in pirte?)
* Can ecm-core, ecm-linux, and simulator be conbined in any way?
* Can server and webportal be conbined in any way?
* Fix the “Vehicle Manufactoring”-page on moped.sics.se

**Server**

* Make sure that the sse computer is accessible from outside? - Currently, it is not
* Make it possible to admin web portal and the trusted server from outside
* Create an sse-account for the trusted server
* Make it possible to upload / delete plugins to our part of wordpress through web interface