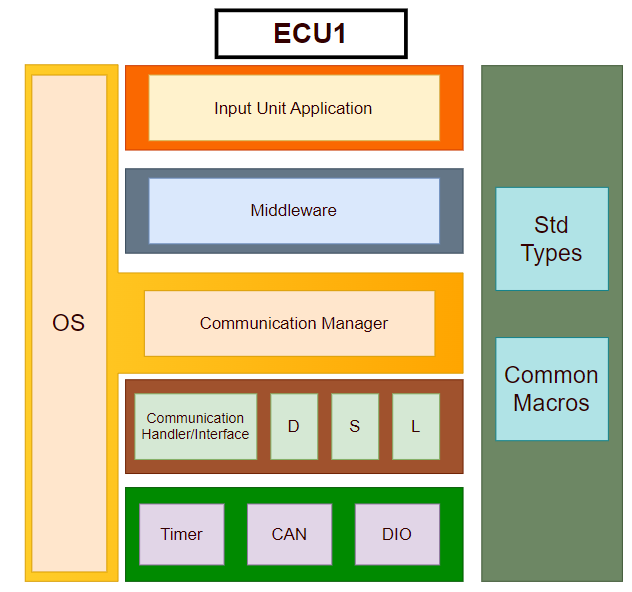
EgFWD

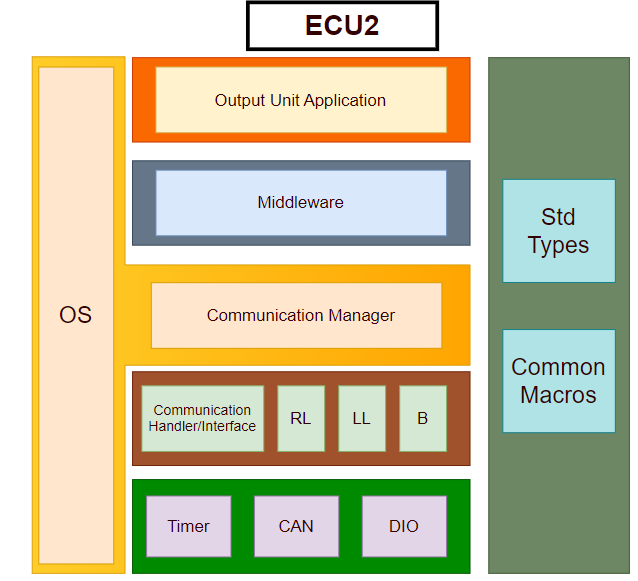
Advanced Embedded Systems

Project (3) | System Design

Automotive Door Control System Design Report

**For ECU 1:**

1. Make the layered architecture
2. Specify ECU components and modules
   * Application: Where system logic and tasks are implemented.
     + main.c
   * Middleware: For routing the system to its desired destination. For example, either to send data from input devices and so being routed to communication manager or else saving this data in EEPROM and so being routed to memory manager.
     + middleware.c | middleware.h
   * Service:
     + RTOS: Handling tasks and operating system as stated in project rubric.
       - All RTOS related files from previous course(RTOS Porting Section | memory management, heap, tasks.c, config.h …etc)
     + Communication(BCM): Used to send Status messages.
       - BCM.c, BCM.h
   * Common: For standard data types and common macros.
     + std\_types.h
     + common\_macros.h
   * HAL:
     + Onboard/ECUAL: containing Door sensor, Light switch, Speed sensor and all external input devices.
       - doorSensor.c, doorSensor.h
       - lightSwitch.c, lightSwitch.h
       - speedSensor.c, speedSensor.h
     + MCAL:
       - Timer
         * timer.c, timer.h
         * timer\_cfg.c, timer\_cfg.h
       - CAN
         * CAN.c, CAN.h
         * CAN\_cfg.c, CAN\_cfg.h
       - DIO
         * DIO.c. DIO.h
         * DIO\_cfg.c, DIO\_cfg.h
3. Provide full detailed APIs for each module as well as a detailed description for the used typedefs
4. Prepare your folder structure according to the previous points

**For ECU 2:**

1. Make the layered architecture
2. Specify ECU components and modules
   1. Application: Where system logic and tasks are implemented.
   2. Middleware: For routing the system to its desired destination.
   3. RTOS: Handling tasks and operating system as stated in project rubric.
   4. Common: For standard data types and common macros.
   5. HAL:
      1. Onboard/ECUAL: containing Right Light , Left Light, Buzzer and all external output devices.
      2. MCAL: containing Timer, CAN and DIO modules.
3. Provide full detailed APIs for each module as well as a detailed description for the used typedefs
4. Prepare your folder structure according to the previous points