AUTOMOTIVE DOOR CONTROL SYSTEM DESIGN

Embedded Systems Advanced Track

Egyptfwd

Submitted by: Salma Zakaria Mohamed Ibrahim

Content

1. Hardware requirements:

Two microcontrollers connected via CAN bus

One Door sensor (D)

One Light switch (L)

One Speed sensor (S)

ECU 1 connected to D, S, and L, all input devices

Two lights, right (RL) and left (LL)

One buzzer (B)

ECU 2 connected to RL, LL, and B, all output devices

2. Software requirements:

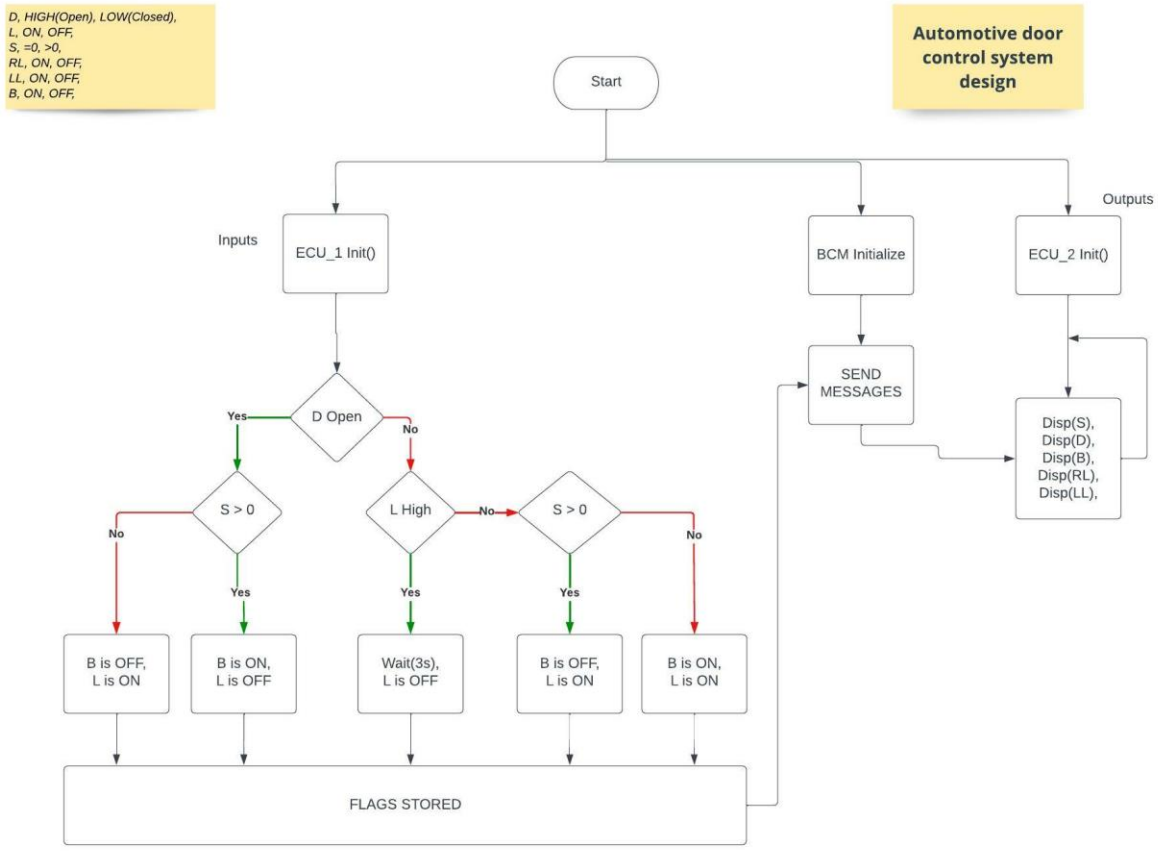
1. ECU 1 will send status messages periodically to ECU 2 through the CAN protocol
2. Status messages will be sent using Basic Communication Module (BCM)
3. Door state message will be sent every 10ms to ECU 2
4. Light switch state message will be sent every 20ms to ECU 2
5. Speed state message will be sent every 5ms to ECU 2
6. Each ECU will have an OS and application SW components
7. If the door is opened while the car is moving → Buzzer ON, Lights OFF
8. If the door is opened while the car is stopped → Buzzer OFF, Lights ON
9. If the door is closed while the lights were ON → Lights are OFF after 3 seconds
10. If the car is moving and the light switch is pressed → Buzzer OFF, Lights ON
11. If the car is stopped and the light switch is pressed → Buzzer ON, Lights ON

3. Components Statuses:

Component	Statuses
D	**HIGH(Open), LOW(Closed)**
S	**=0, >0**
RL	**ON, OFF**
LL	**ON, OFF**
B	**ON, OFF**

4. Block Diagram

View Only Link on *Lucid.app*



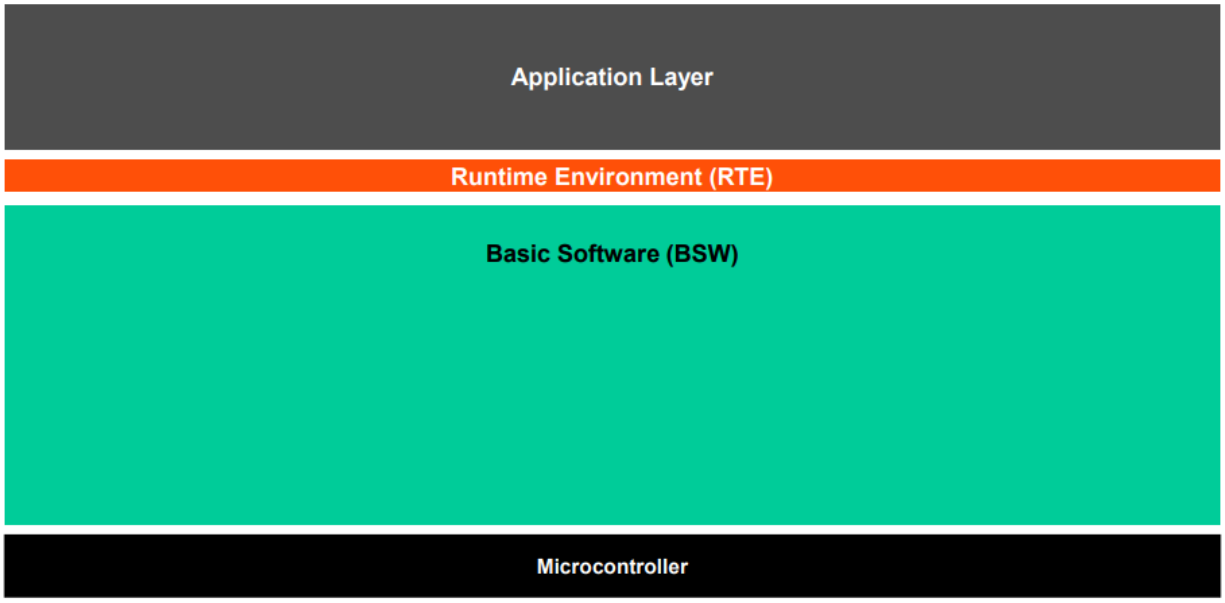
5. Static Design

6. ECU 1

7. Layered Architecture

AUTOSAR Layered Architecture

The AUTOSAR Architecture distinguishes on the highest abstraction level between three software layers: Application, Runtime Environment and Basic Software which run on a Microcontroller.



Basic Software Layer (BSW):

The Microcontroller Abstraction Layer is the lowest software layer of the Basic Software. It contains internal drivers, which are software modules with direct access to the µC and internal peripherals.

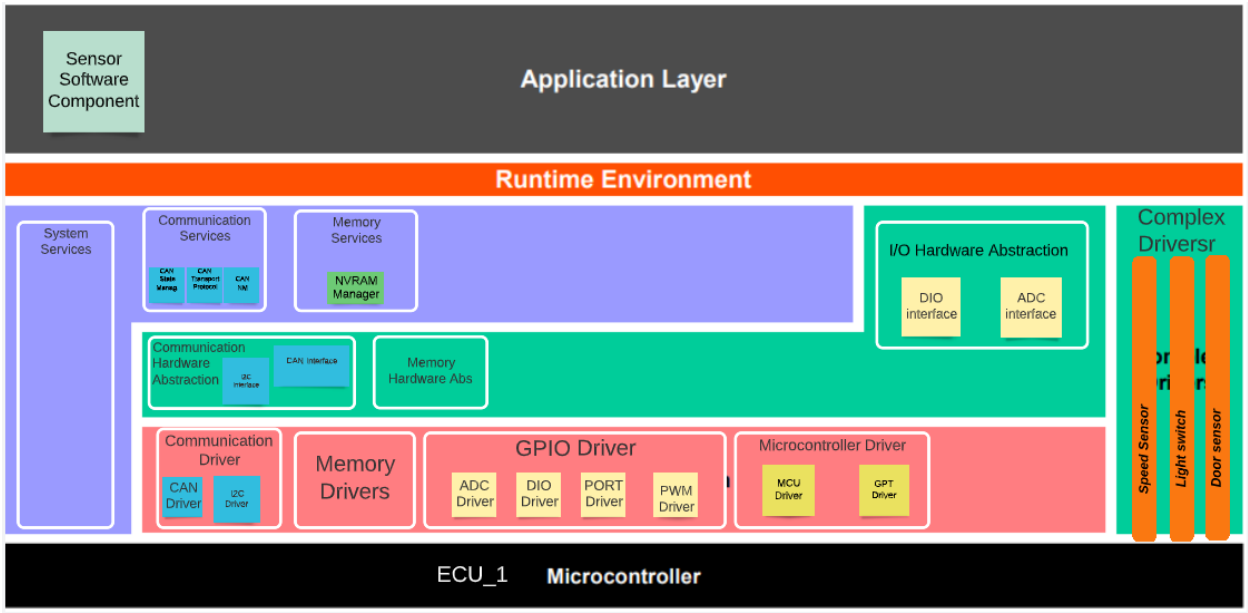
The ECU Abstraction Layer interfaces the drivers of the Microcontroller Abstraction Layer. It also contains drivers for **external devices**. It offers an API for access to peripherals and devices regardless of their location (µC internal/external) and their connection to the µC (port pins, type of interface).

The Complex Drivers Layer spans from the hardware to the RTE.

The Services Layer is the highest layer of the Basic Software which also applies for its relevance for the application software: while access to I/O signals is covered by the ECU Abstraction Layer, the Services Layer offers: ¬ Operating system functionality ¬ Vehicle network communication and management services ¬ Memory services (NVRAM management) ¬

Diagnostic Services (including UDS communication, error memory and fault treatment) ¬ ECU state management, mode management ¬ Logical and temporal program flow monitoring (Wdg manager)

ECU 1 Layered Application.



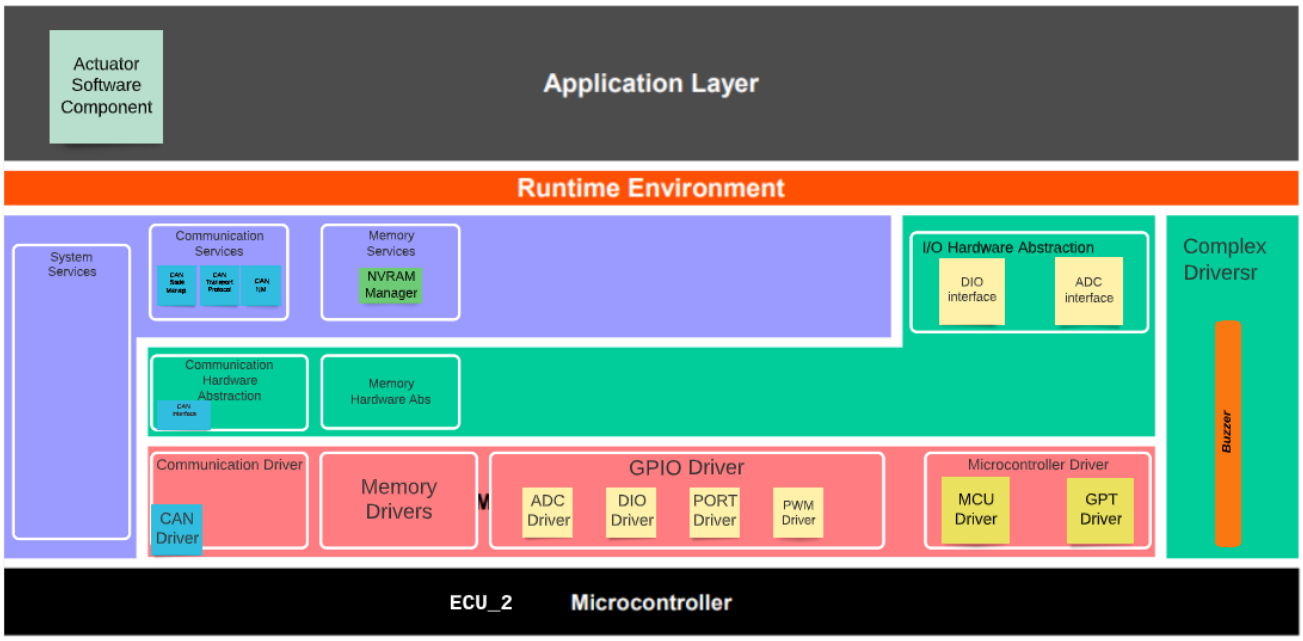
4.1.1. Components and Modules:

- Memory Manager Module
- CAN Module
- GPIO Module
- Timer Module
- Speed Sensor Module
- Light Switch Module (Latch Button)
- Door Sensor

4.2. ECU 2

4.2.1. Layered Architecture

Assuming RL and LL are LEDs.



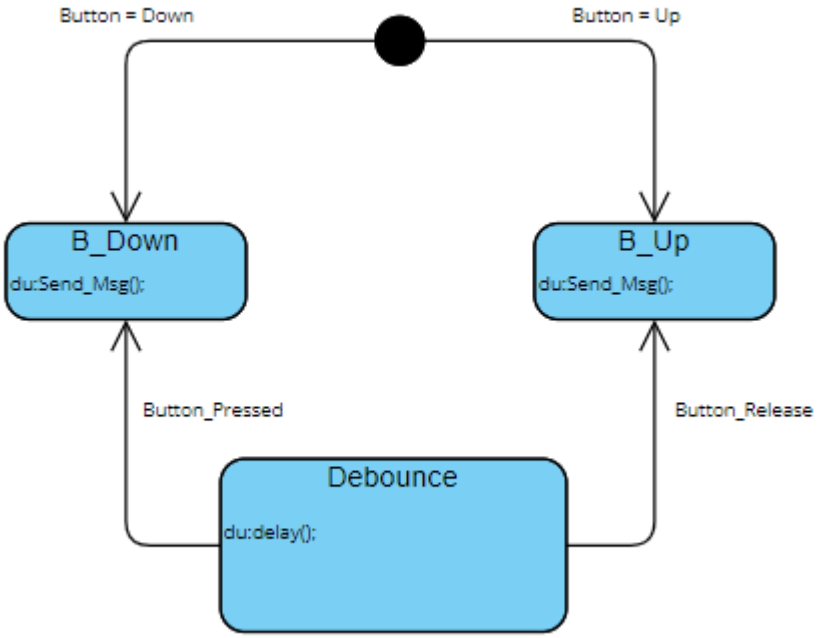
4.2.2. Components and Modules

- Memory Manager Module
- CAN Module
- GPIO Module
- Timer Module
- Buzzer Module

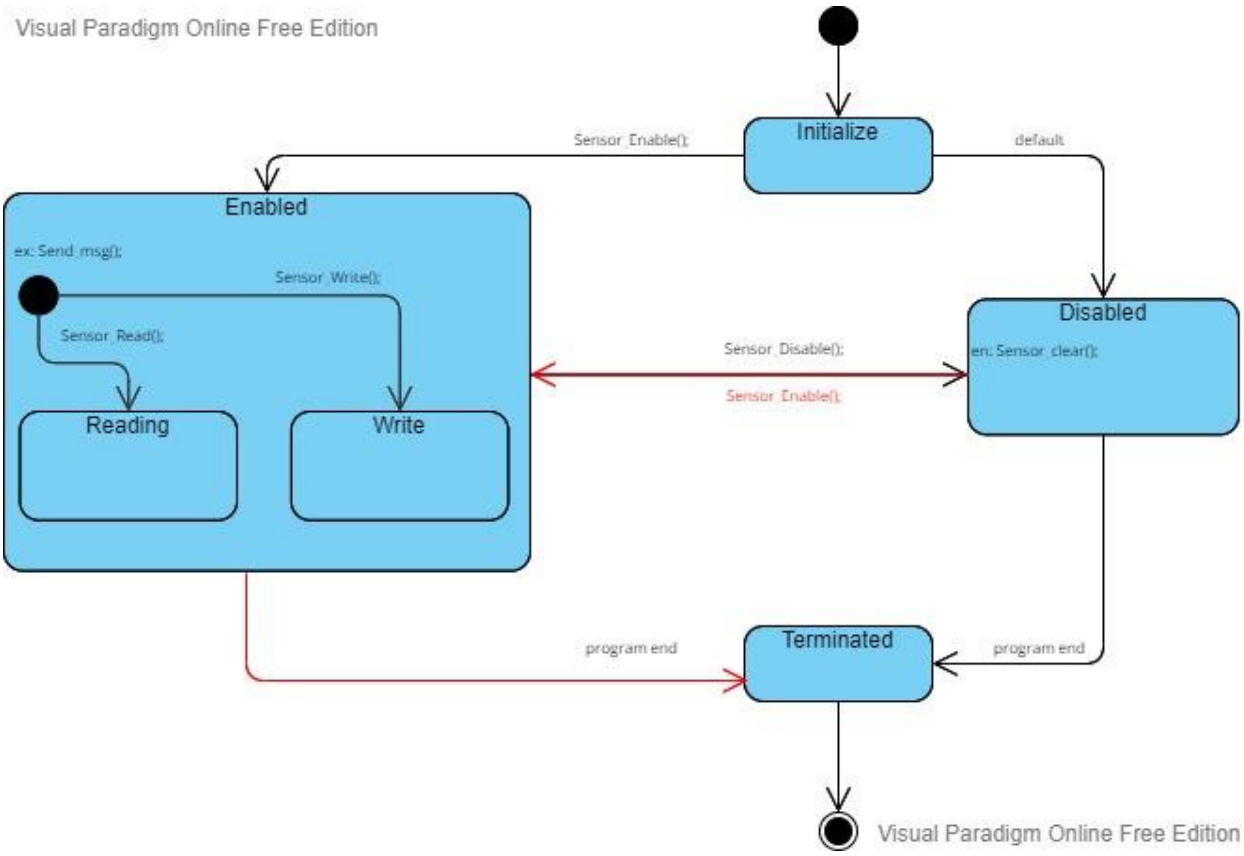
8. Dynamic Design

8.1. State Machine Component

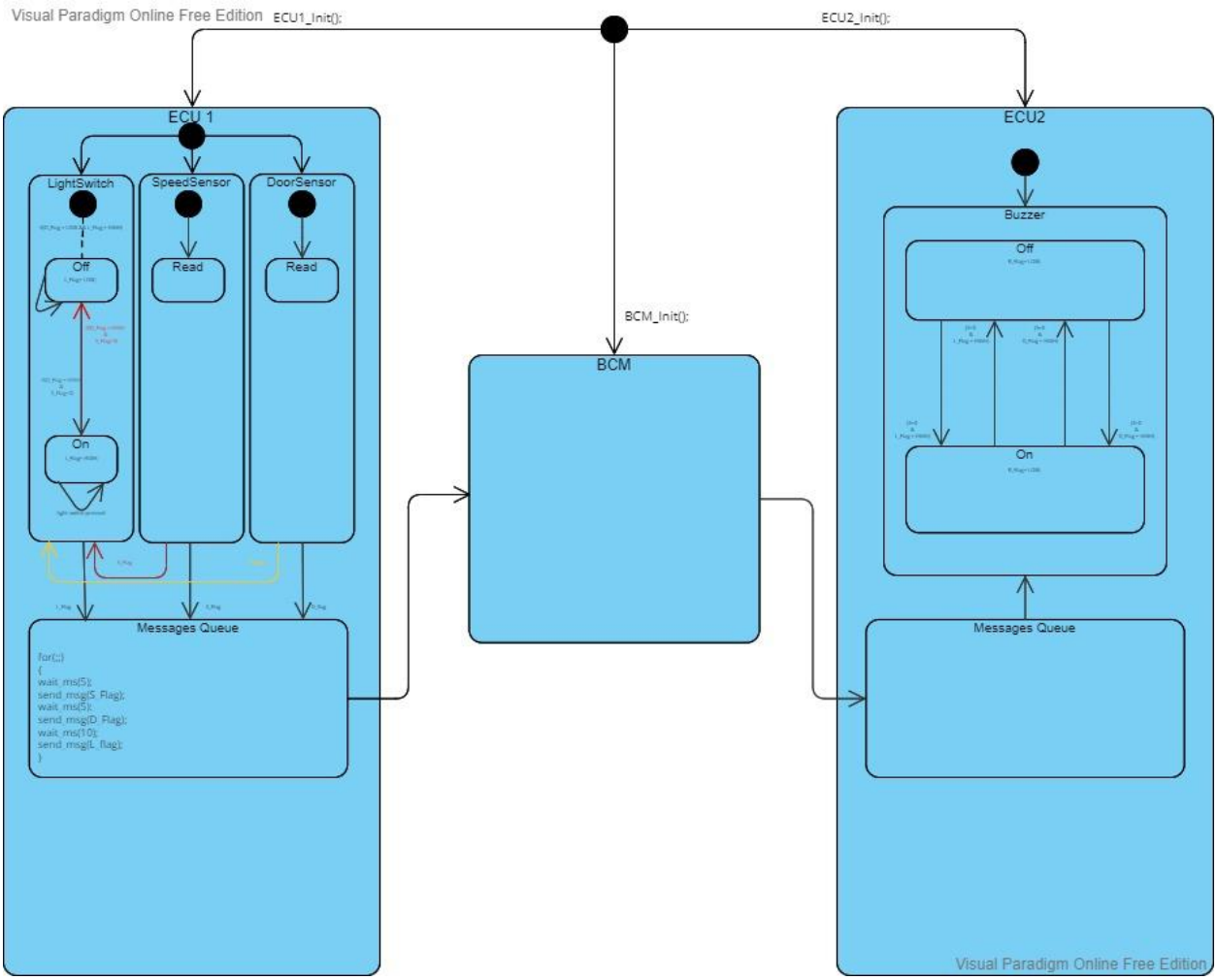
Push Button



Speed Sensor



8.2. State Machine Operation



8.3. Sequence Diagram