

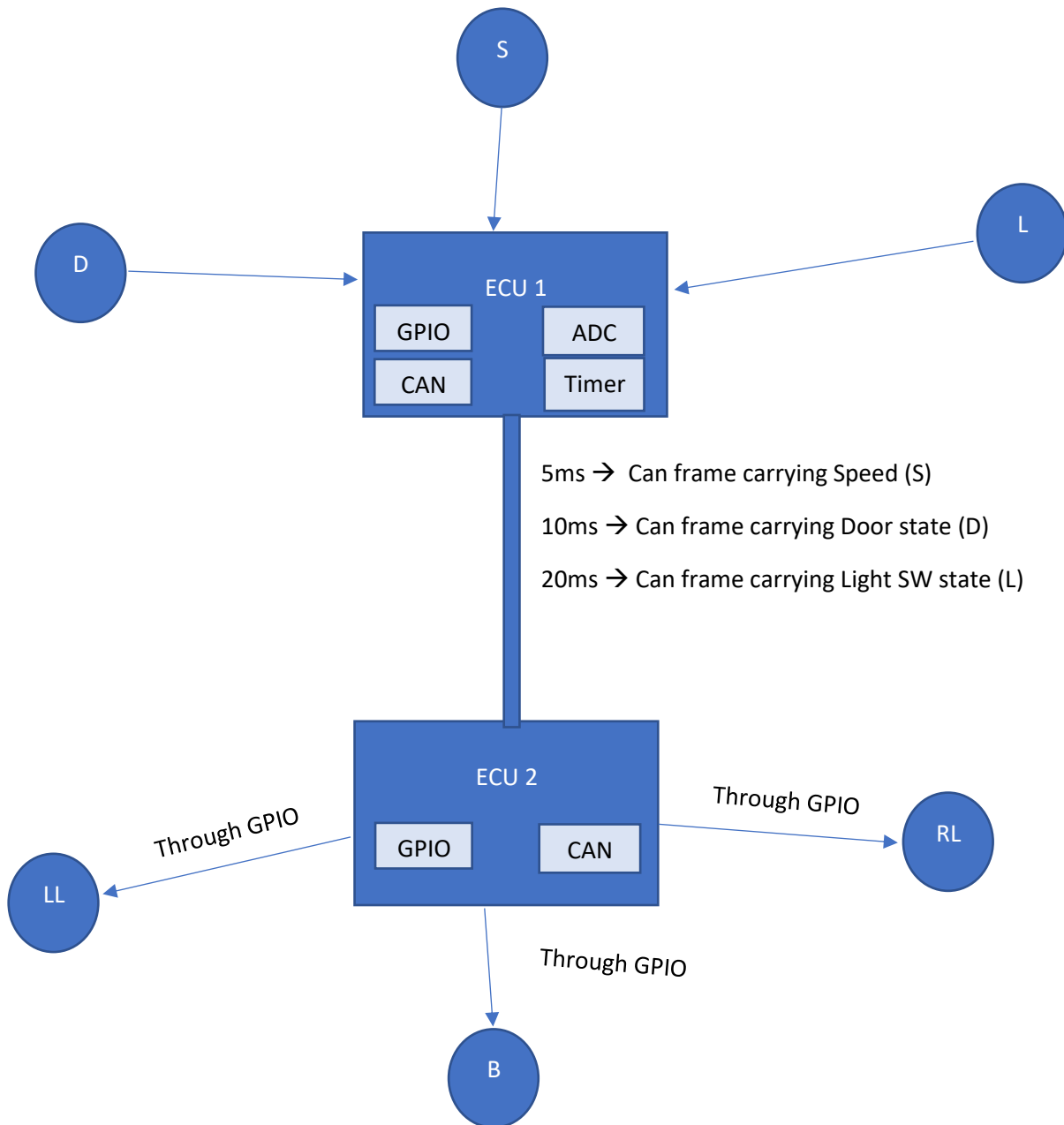
Automotive door control system design

egFWD – Embedded Systems

Advanced Track

By: Mohamed Elsayed

1-System schematic



RL: Right Light

LL: Left Light

B: Buzzer

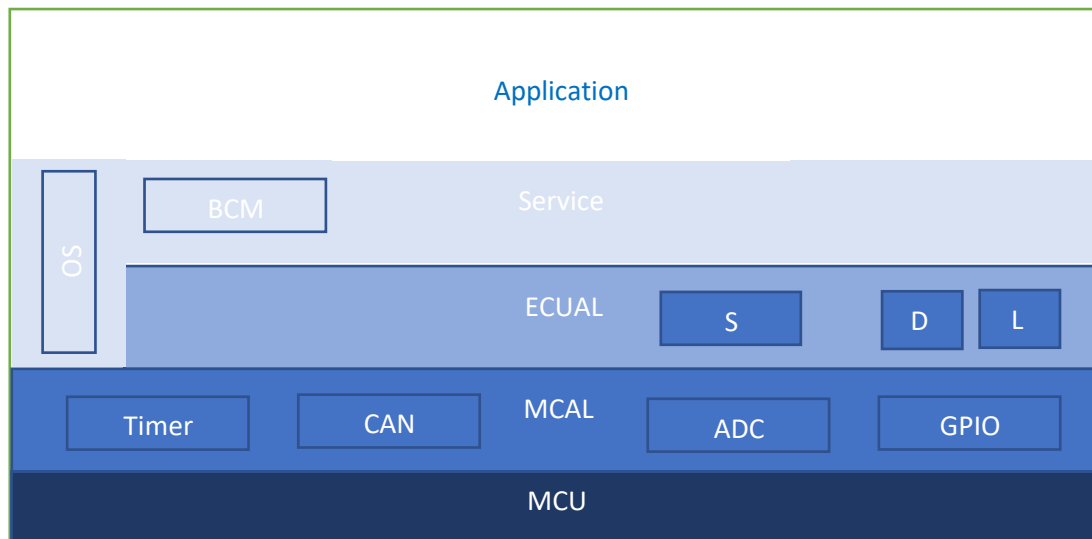
S: speed sensor

D: Door sensor

L: Light switch

2- Static design analysis

ECU 1:



System will consist of different SWCs like (Timer, DIO, Button, LED, Utilities, APP) in three different Layers:

- MCAL layer for (GPIO, ADC, Timer, CAN)

- ECUAL layer for (Speed sensor, Door Sensor, Light switch)

- APPLICATION layer for our app

MCAL Layer

This layer consists of four modules:

1-GPIO Module

This module will be responsible for interfacing between I/O devices and MCU.

This module will consist of the following APIs:

```
/*Initialize GPIO direction*/  
void GPIO_init(uint8_t portnumber, uint8_t pinnumber, uint8_t direction);  
/*write data to GPIO*/  
void GPIO_write(uint8_t portnumber, uint8_t pinnumber, uint8_t value);  
/*toggle GPIO*/  
void GPIO_toggle(uint8_t portnumber, uint8_t pinnumber);  
/*Read GPIO*/  
void GPIO_read(uint8_t portnumber, uint8_t pinnumber, uint8_t* value);
```

2-Timer Module

This module will be responsible for Timing delays and functions in the system.

This module will consist of the following APIs:

```
/*Initialize Timer*/  
void TIMER_init(void);  
/* Turn on Timer*/  
void TIMER_on(void);  
/*Turn off Timer */  
void TIMER_off(void);
```

3-ADC Module

This module will be responsible for reading values analog values from sensors.

This module will consist of the following APIs:

```
/*Initialize ADC*/  
void ADC_init(void);  
  
/*Read ADC*/  
uint16_t ADC_read(uint8_t Channelnumber);
```

4-CAN Module

This module will be responsible for communications with other ECUs.

This module will consist of the following APIs:

```
/*Initialize CAN*/  
void CAN_init(void);  
  
/*Initialize CAN*/  
void CAN_send(uint8_t Data);
```

ECUAL Layer

This layer consists of three modules:

1-Speed Sensor Module

This module will be responsible for reading the speed sensor.

This module will consist of the following APIs:

```
/*Initialize Speed sensor*/  
void Speedsensor_init(uint8_t sensorPort,uint8_t sensorPin);  
/*Read Speed sensor*/  
uint16_t Speed_read(void);
```

2-Door Sensor Module

This module will be responsible for reading crosswalk pushbutton.

This module will consist of the following APIs:

```
/*Initialize Door State*/  
void Door_Sensor_init(uint8_t sensorPort,uint8_t sensorPin);  
/*Read Door State*/  
uint8_t Door_state_read(void);
```

3-Light Switch Module

This module will be responsible for reading crosswalk pushbutton.

This module will consist of the following APIs:

```
/*Initialize Light switch*/  
void Light_Switch_init(uint8_t switchPort,uint8_t switchPin);  
/*Read Light switch State*/  
uint8_t Light_Switch_read(void);
```

APPLICATION Layer

This layer consists of only one modules:

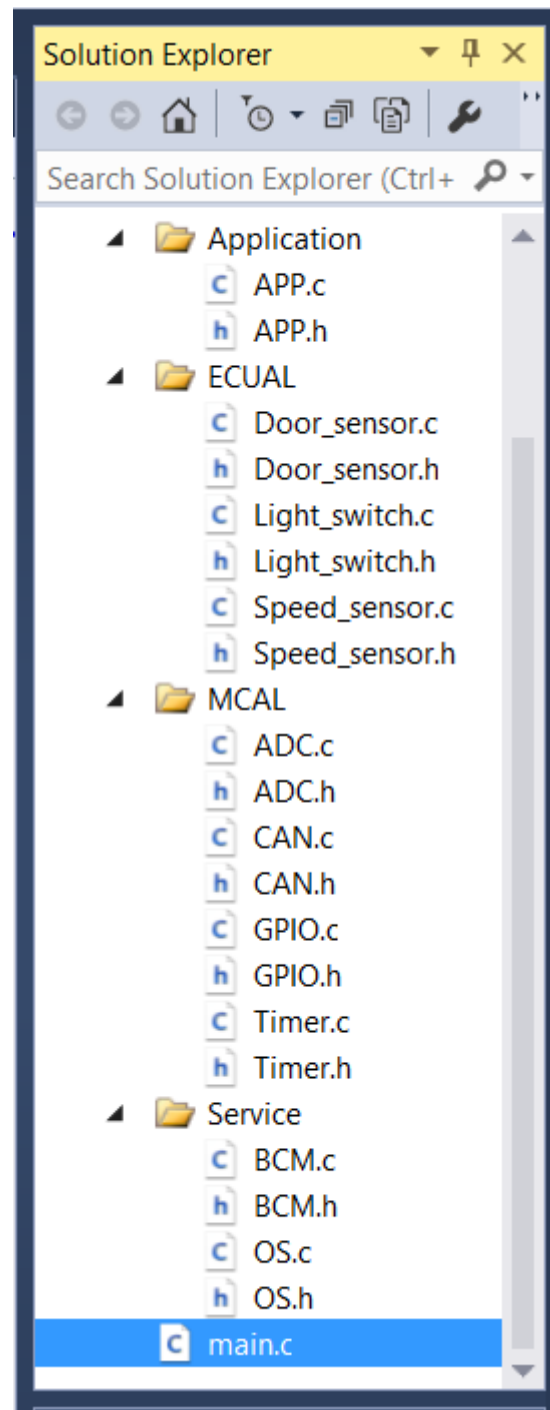
APP Module

This module will be responsible for the flow of the project.

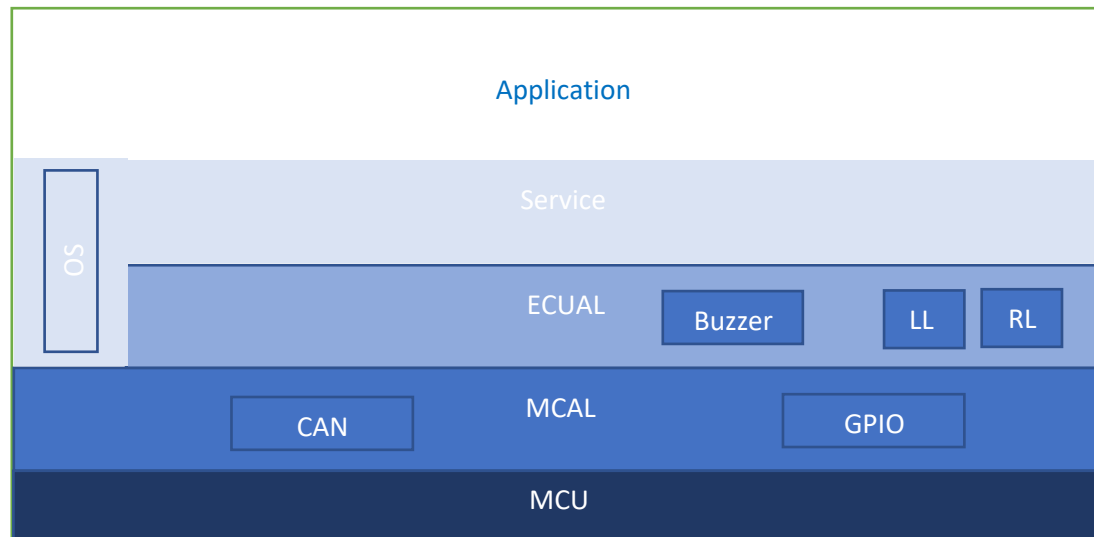
This module will consist of the following APIs:

```
/*Initialize ECUAL Modules in the APP*/  
void APP_init();  
  
/*Start the APP*/  
void APP_start();
```

Folder Structure



ECU 2:



System will consist of different SWCs like (GPIO, CAN, Buzzer, Light, OS, APP) in four different Layers:

- Service layer for (OS)
- MCAL layer for (GPIO, CAN)
- ECUAL layer for (Buzzer, Light)
- APPLICATION layer for our app

MCAL Layer

This layer consists of four modules:

1-GPIO Module

This module will be responsible for interfacing between I/O devices and MCU.

This module will consist of the following APIs:

```
/*Initialize GPIO direction*/  
void GPIO_init(uint8_t portnumber, uint8_t pinnumber, uint8_t direction);  
/*write data to GPIO*/  
void GPIO_write(uint8_t portnumber, uint8_t pinnumber, uint8_t value);  
/*toggle GPIO*/  
void GPIO_toggle(uint8_t portnumber, uint8_t pinnumber);  
/*Read GPIO*/  
void GPIO_read(uint8_t portnumber, uint8_t pinnumber, uint8_t* value);
```

2-CAN Module

This module will be responsible for communications with other ECUs.

This module will consist of the following APIs:

```
/*Initialize CAN*/  
void CAN_init(void);  
  
/*Initialize CAN*/  
uint8_t CAN_receive(void);
```

ECUAL Layer

This layer consists of three modules:

1-Buzzer Module

This module will be responsible for reading the speed sensor.

This module will consist of the following APIs:

```
/*Initialize Speed sensor*/  
void Buzzer_init(uint8_t buzzerPort,uint8_t buzzerPin);  
/*Turn Buzzer ON*/  
void Buzzer_ON(uint8_t buzzerPort,uint8_t buzzerPin);  
/*Turn Buzzer OFF*/  
void Buzzer_OFF(uint8_t buzzerPort,uint8_t buzzerPin);
```

2-Light Module

This module will be responsible for reading crosswalk pushbutton.

This module will consist of the following APIs:

```
/*Initialize Light switch*/  
void Light_init(uint8_t lightPort,uint8_t lightPin);  
/*Turn Light ON*/  
void Light_ON(uint8_t lightPort,uint8_t lightPin);  
/*Turn Light OFF*/  
void Light_OFF(uint8_t lightPort,uint8_t lightPin);
```

APPLICATION Layer

This layer consists of only one modules:

APP Module

This module will be responsible for the flow of the project.

This module will consist of the following APIs:

```
/*Initialize ECUAL Modules in the APP*/  
void APP_init();  
  
/*Start the APP*/  
void APP_start();
```


Folder Structure

