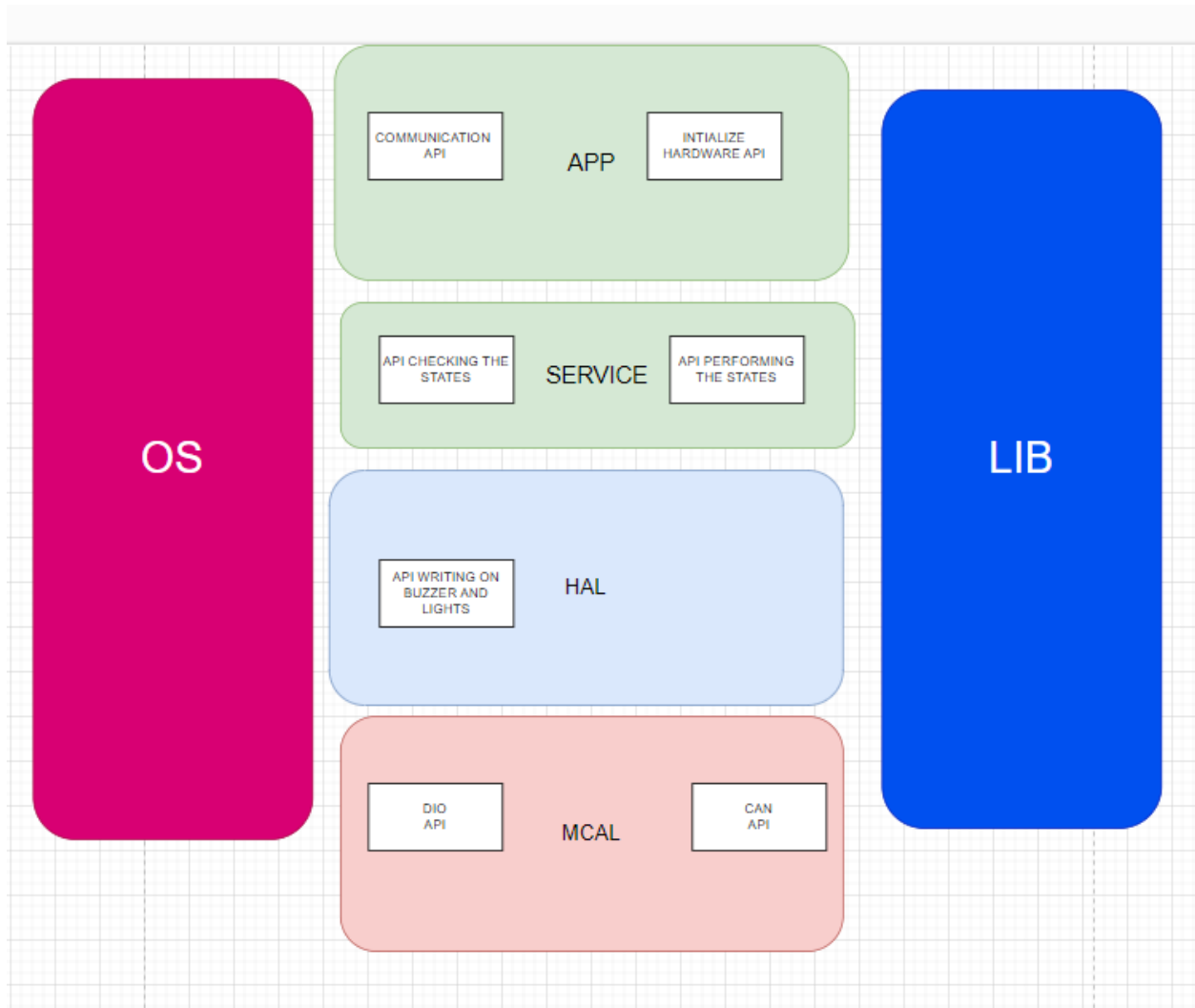
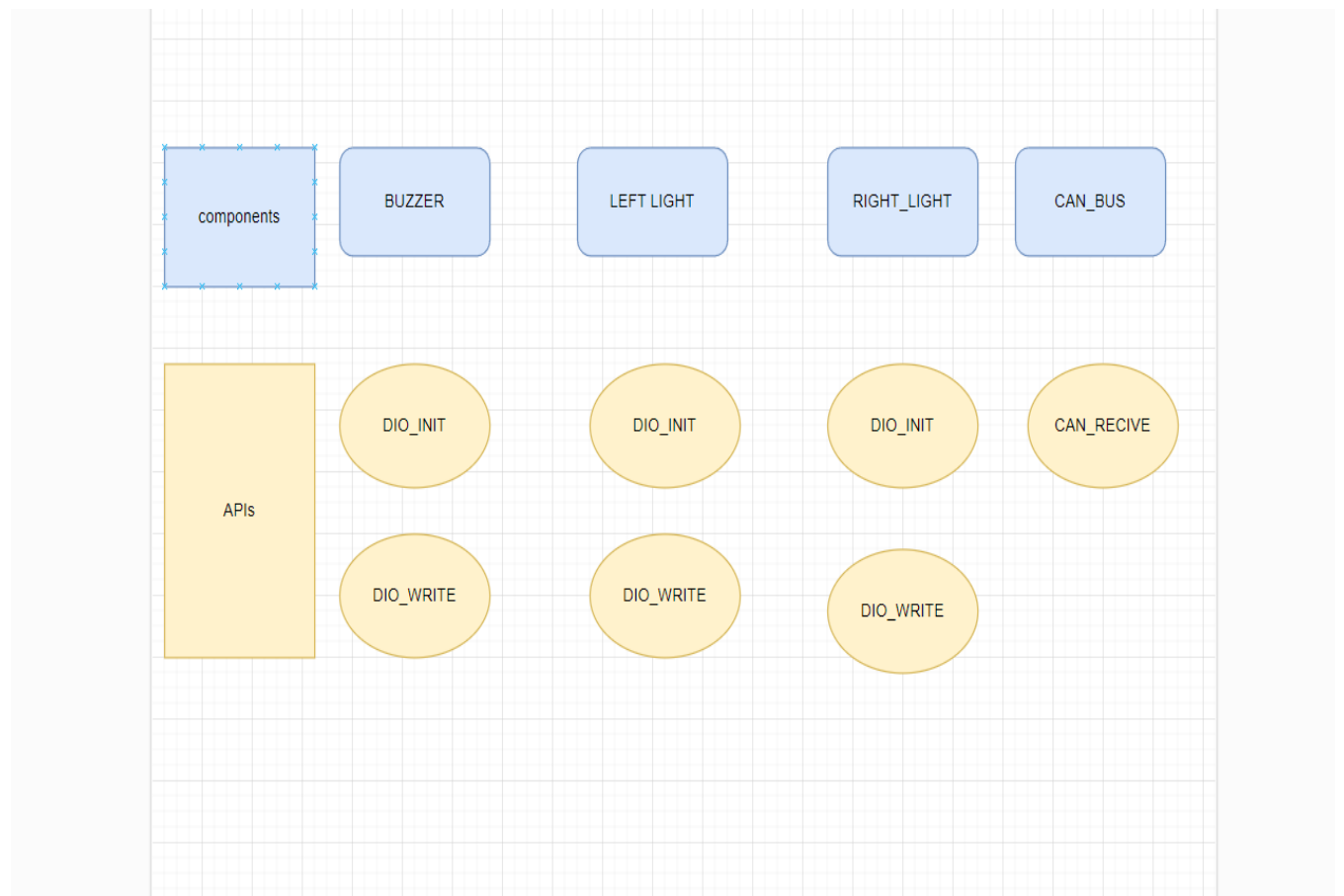


## Static design for ECU2

### Layered architecture

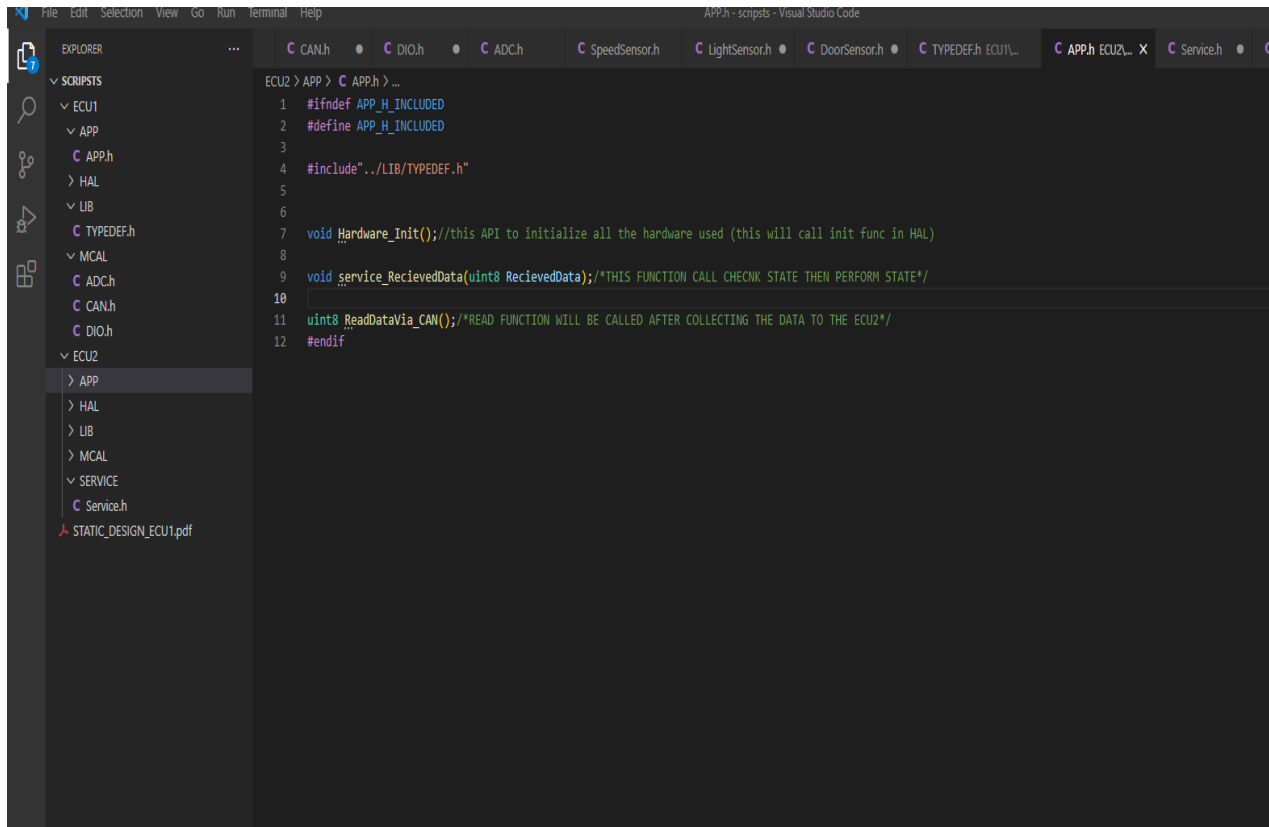


### COMPONENTS AND MODULES



BUZZER ,LIGHTS,CAN\_BUS

# APP ,APIs



## Void Hardware\_Init()

To initialize all the hardware I have bu calling all init function in HAL layer

## Void Service\_RecievedData(uint8 ReciecvdData)

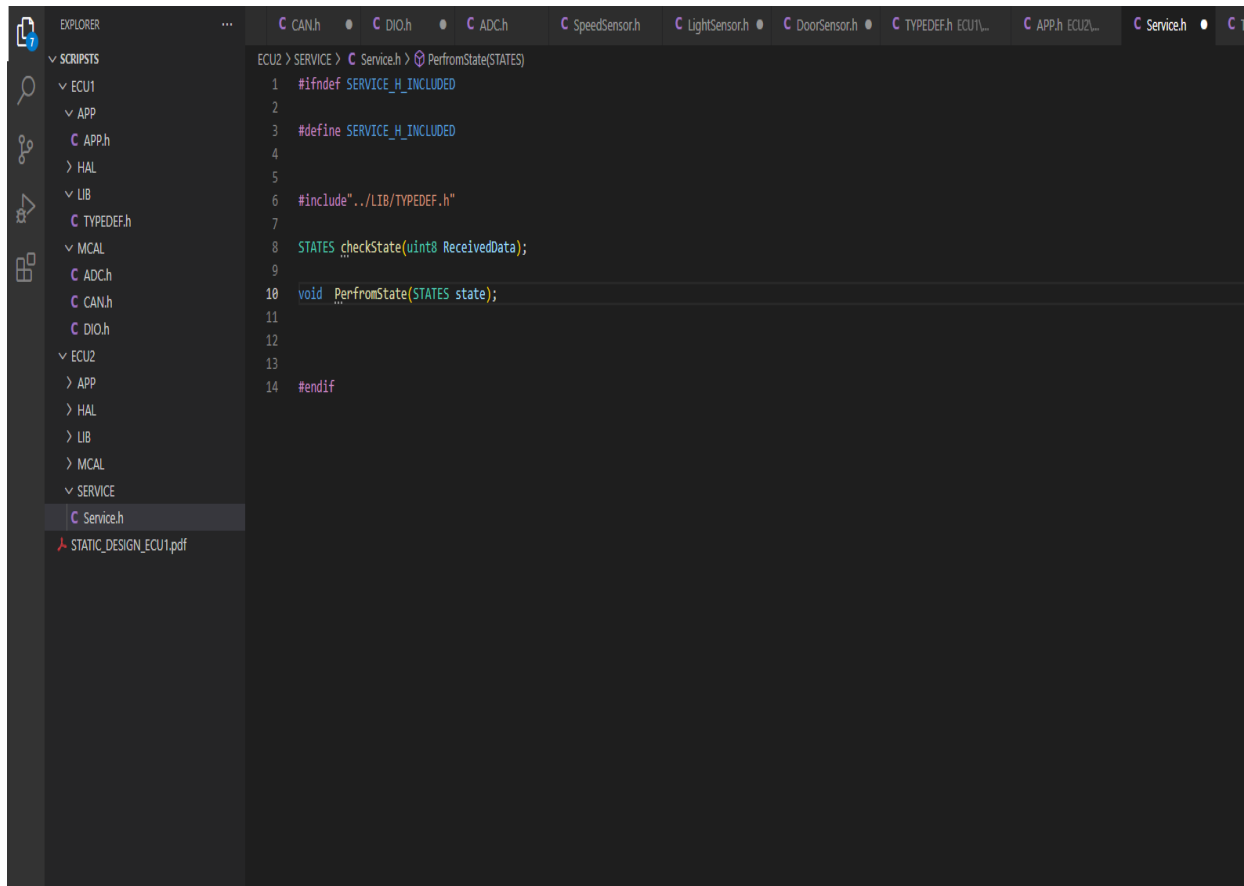
This func call

STATES checkstates());Then

void perform\_states(STATES state); from service layer

So the received data be recognized and performing the actions according this checkState

## SERVICE APIs



STATES checkState(uint8 ReceivedData);

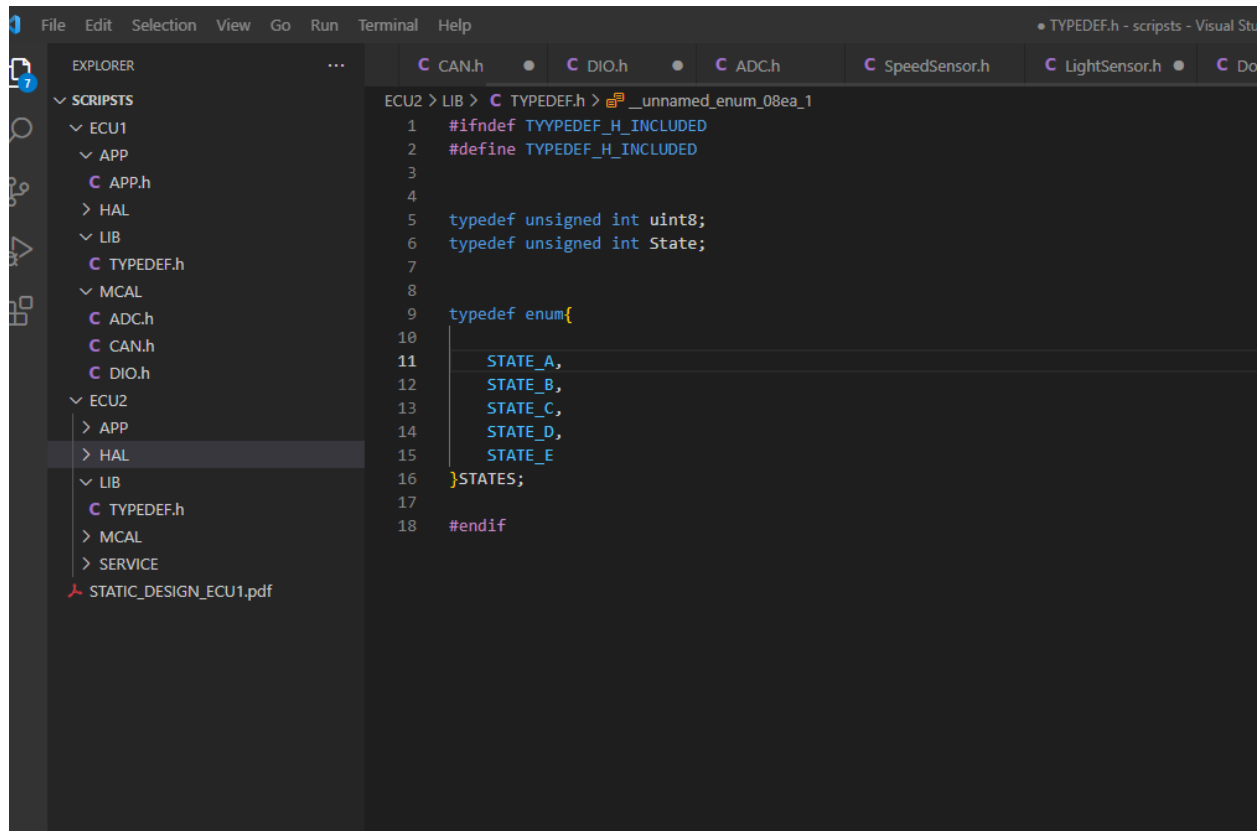
This function check the received data from can and return if the STATE was(A,B,C,D,E,F)

Void PerformStates(STATES state);

This function perform the state as described in the rubric

As it call buzzer functions and lights functions

## TYPEDEFS

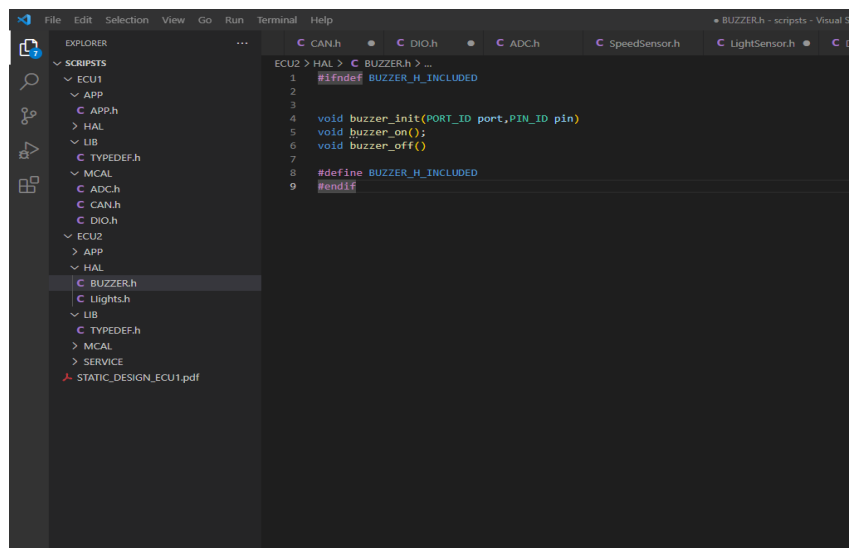


```
File Edit Selection View Go Run Terminal Help
• TYPEDEF.h - scripsts - Visual Studio Code

EXPLORER
  SCRIPSTS
  ECU1
  APP
  APP.h
  HAL
  LIB
  TYPEDEF.h
  MCAL
  ADC.h
  CAN.h
  DIO.h
  ECU2
  APP
  HAL
  LIB
  TYPEDEF.h
  MCAL
  SERVICE
  STATIC_DESIGN_ECU1.pdf

ECU2 > LIB > C TYPEDEF.h > _unnamed_enum_08ea_1
1  #ifndef TYPEDEF_H_INCLUDED
2  #define TYPEDEF_H_INCLUDED
3
4
5  typedef unsigned int uint8;
6  typedef unsigned int State;
7
8
9  typedef enum{
10
11     STATE_A,
12     STATE_B,
13     STATE_C,
14     STATE_D,
15     STATE_E
16 }STATES;
17
18 #endif
```

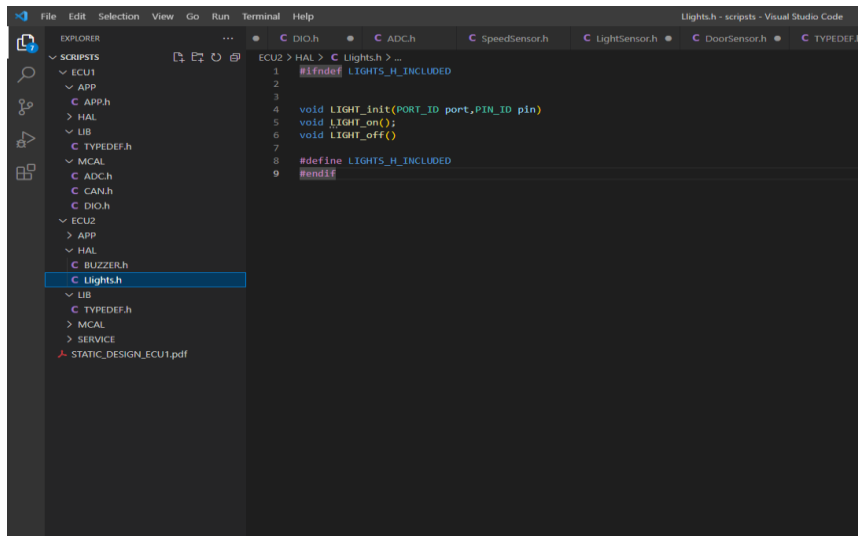
## HAL APIS



```
File Edit Selection View Go Run Terminal Help
• BUZZER.h - scripsts - Visual Studio Code

EXPLORER
  SCRIPSTS
  ECU1
  APP
  APP.h
  HAL
  LIB
  TYPEDEF.h
  MCAL
  ADC.h
  CAN.h
  DIO.h
  ECU2
  APP
  HAL
  BUZZER.h
  LIGHT.h
  LIB
  TYPEDEF.h
  MCAL
  SERVICE
  STATIC_DESIGN_ECU1.pdf

ECU2 > HAL > C BUZZER.h > ...
1  #ifndef BUZZER_H_INCLUDED
2
3
4  void buzzer_init(PORT_ID port, PIN_ID pin)
5  void buzzer_on();
6  void buzzer_off();
7
8  #define BUZZER_H_INCLUDED
9  #endif
```



## MCAL APIs

```
1 typedef enum {
2     LOW,
3     HIGH
4 }LEVEL_DATA;
5
6 typedef enum {
7     PIN0,
8     PIN1,
9     PIN2,
10    PIN3,
11    PIN4,
12    PIN5,
13    PIN6,
14    PIN7,
15 }PIN_ID;
16
17 typedef enum {
18     PORT_A,
19     PORT_B,
20     PORT_C,
21     PORT_D,
22 }PORT_ID;
23
24 typedef struct DIO_PIN_CONFIG
25 {
26     /* data */
27     PIN_ID pin;
28     PORT_ID port;
29     MODE mode;
30 };
31
32 void DIO_Init_PIN(DIO_PIN_CONFIG pin_struct);
33 State DIO_ReadChannel(PORT_ID port,PIN_ID pin);
34 void DIO_WriteChannel(PORT_ID port,PIN_ID pin);
35
36 #define DIO_H_INCLUDED
37
38 #endif
```

