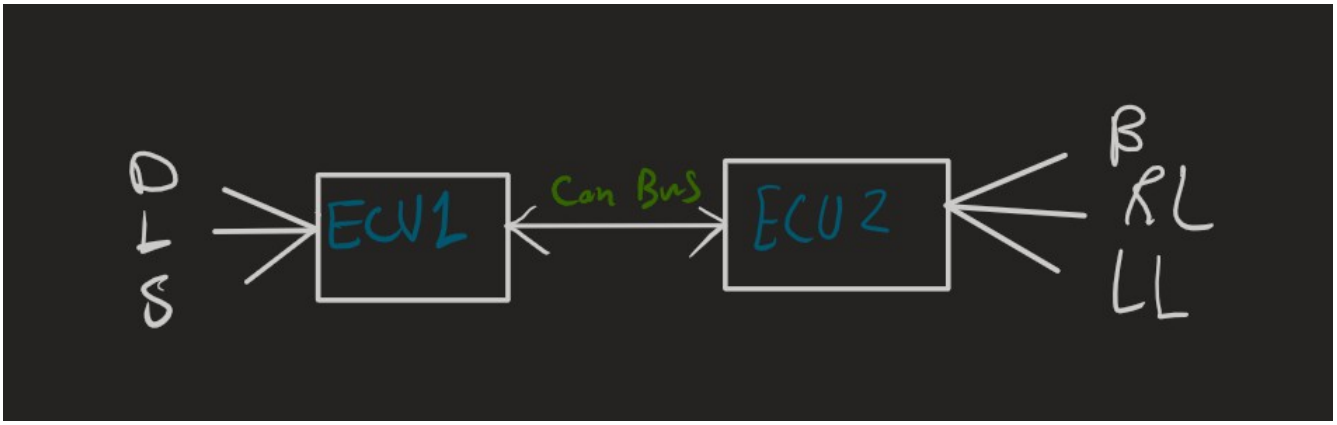
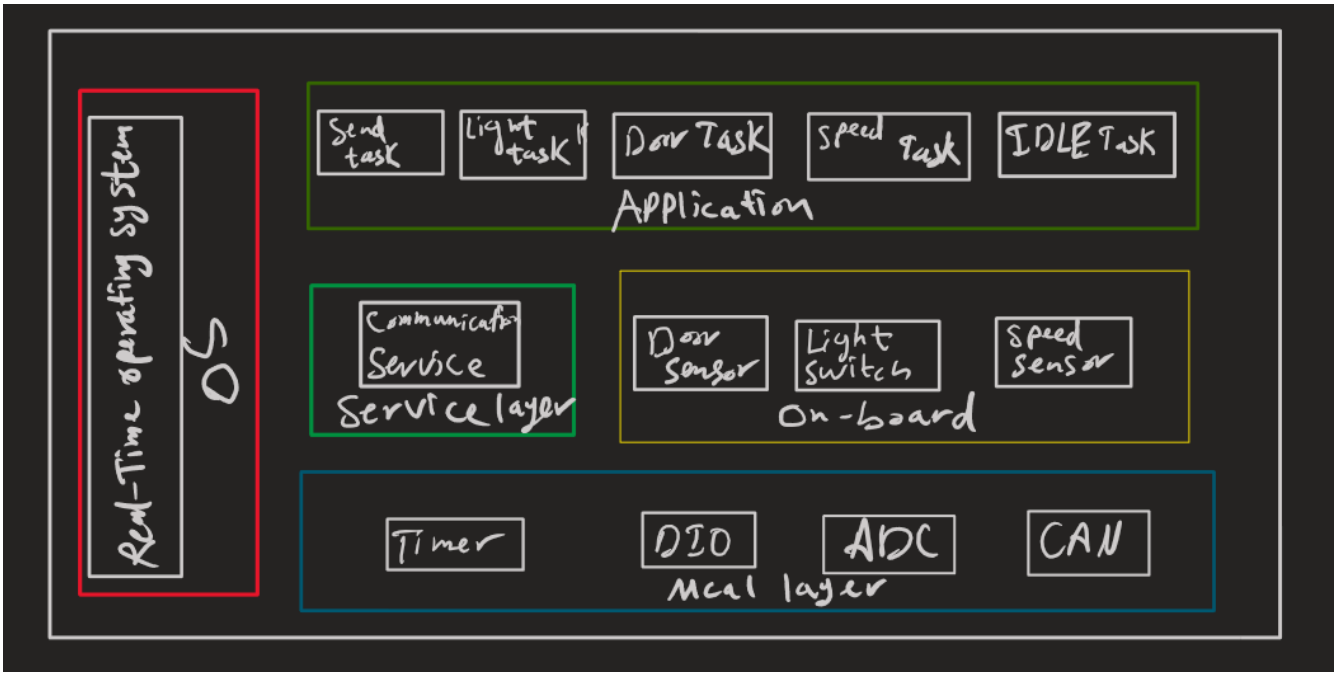
Hardware Connections:

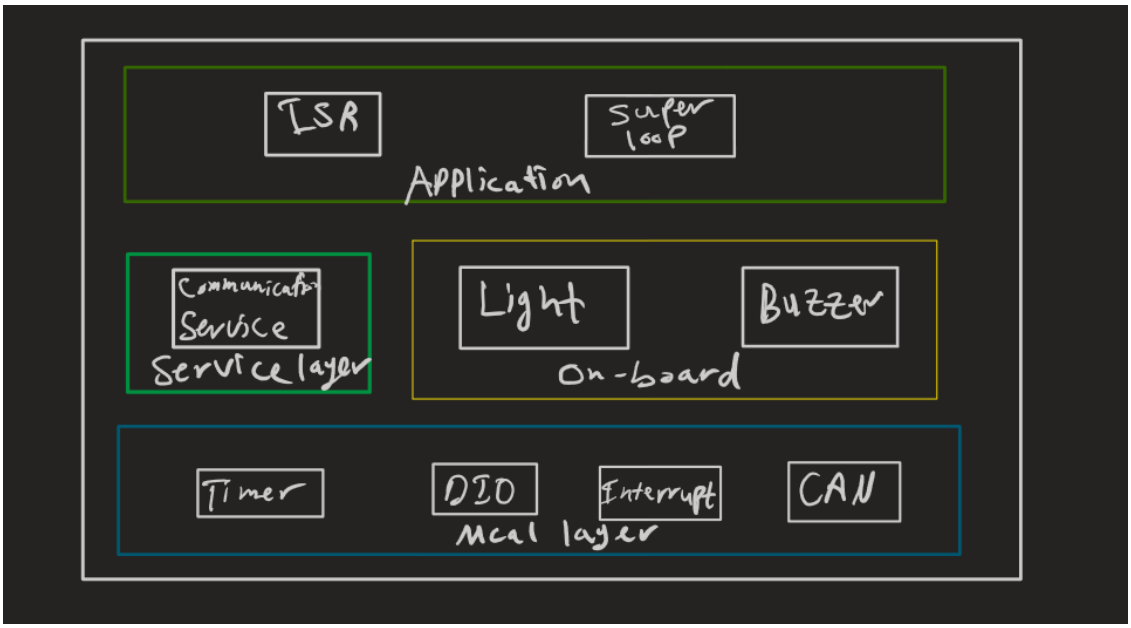


Static Desing:

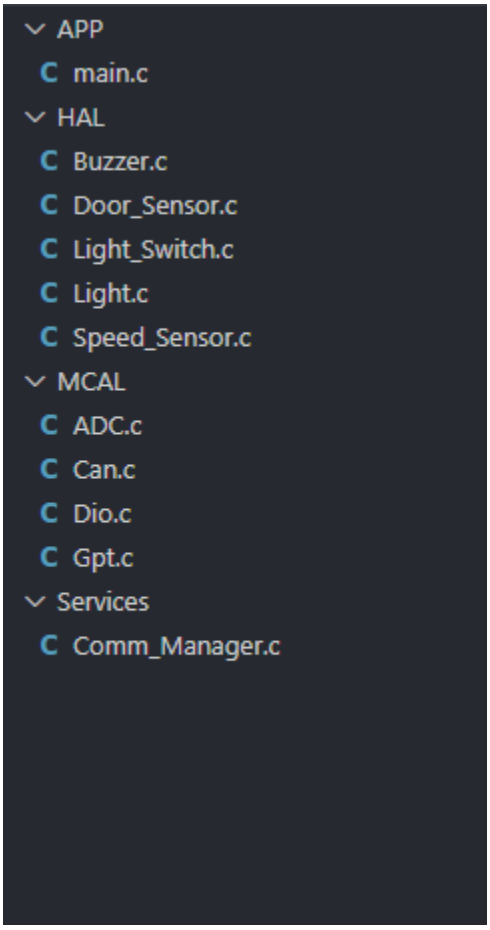
ECU1 runs Operating System:



ECU2:



Folder Structure:



* GPT (General Purpose timer) Module:

Required APIs:

```
void GPT_init(GPT_Config_Ptr* config);
void GPT_Start(uint32_t ticks);
void GPT_Stop(void);
void GPT_Delay(uint32_t delay);
```

* DIO (Digital Input Output) module:
Required APIs:

```
void DIO_init(DIO_Config_Ptr* config);
void DIO_Write(DIO_pin pin , uint8_t value);
void DIO_Read(DIO_pin pin , uint8_t* value);
void DIO_Toggle(DIO_pin pin);
```

* CAN Module:
Required APIs:

```
void CAN_init(CAN_Config_Ptr* config);
void CAN_Send(uint32_t* data);
void CAN_Receive(uint32_t* data);
```

* ADC (Analoge to Digital Converter) Module:
Required APIs:

```
void ADC_init(ADC_Config_Ptr* config);
void ADC_Read(ADC_pin pin , uint8_t* value);
```

* Door Sensor Required APIs:

```
void Door_Sensor_init(Door_Sensor_Config_Ptr* config);
void Door_Sensor_Read(Door_Sensor_pin pin , uint8_t* value);
```

* Light Switch Required APIs:

```
void Light_Switch_init(Light_Switch_Config_Ptr* config);
void Light_Switch_Read(Light_Switch_pin pin , uint8_t* value);
```

* Speed Sensor Required APIs:

```
void Speed_Sensor_init(Speed_Sensor_Config_Ptr* config);
void Speed_Sensor_Read(Speed_Sensor_pin pin , uint8_t* value);
```

* Light Required APIs:

```
void Light_init(Light_Config_Ptr* config);
void Light_On(Light_pin pin);
void Light_Off(Light_pin pin);
```

* Buzzer Required APIs:

```
void Buzzer_init(Buzzer_Config_Ptr* config);
void Buzzer_On(Buzzer_pin pin);
void Buzzer_Off(Buzzer_pin pin);
```