 

# Static Design for Door Control System

**Name: Omar Maher Abd El-Zaher**

**Email: omarmahermaher95@gmail.com**

**System Schematic (Block Diagram)**

Diagram

Description automatically generated

## ECU 1

1. **Layered Architecture**

Chart, bar chart

Description automatically generated

1. **ECU Components & Modules**

#### ECU Interface:

1. CAN Communication
2. Switch
3. Speed Sensor
4. Door Sensor

#### Interface Modules:

1. GPIO Module
2. GPT Module
3. CAN Module
4. ADC Module

### Modules APIs

|  |  |  |  |
| --- | --- | --- | --- |
| **Layer** | **Module** | **API** | **Description** |
| Application Layer | Main Application | Void DoorSensorTask(void) | **Sync/Async:** Synchronous  **Reentrancy**: Non-Reentrant  **Args**: Void  **Return**: Void  **Description**: Get Sensor Data |
| Void LightSwitchTask(void) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: Void  **Return**: Void  **Description**: Control Light Switch |
| Void SpeedSensorTask(void) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: Void  **Return**: Void  **Description**: Get Sensor Data |

|  |  |  |  |
| --- | --- | --- | --- |
| **Layer** | **Module** | **API** | **Description** |
| Service Layer | Sensor Manager | Uint8 ReadSensorData(Sensor\_t Sensor\_ID) | **Sync/Async:** Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Sensor\_ID**: Type of Sensor to be read  **Return**: Byte of Read Data  **Description**: Get Sensor Data |
| Status\_t InitSensor(Sensor\_t Sensor\_ID, Sensor\_Cfg Cfg) | **Sync/Async:** Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Sensor\_ID**: Type of Sensor to be read  **Sensor\_Cfg**: Sensor Configuration Structure  **Return**: Success/Fail Status  **Description**: Initialize Sensor |
| Comm. Manager | Status\_t TransmiData(uint64 Data) | **Sync/Async**: Asynchronous  **Reentrancy**: Non-Reentrant  **Args**: **Data**: to be sent on CAN bus  **Return**: Success/Fail Status  **Description**: Manages Data Transmission over Comm. Protocols |
| Status\_t ReceiveData(uint64 \*Buffer) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Buffer**: Buffer for received data  **Return**: Succeed/Fail Status  **Description**: Receive Transmitted Data |
| Status\_t InitCommManager(Cmanager\_Cfg Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Data**: to be sent on CAN bus  **Return**: Success/Fail Status  **Description**: Initialize Communication Protocol |

|  |  |  |  |
| --- | --- | --- | --- |
| Layer | Module | API | **Description** |
| On-Board Layer | Light Switch | Status\_t initSwitch(Switch\_Cfg Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg** : Light Config Structure  **Return**: Success/Fail Status  **Description:** Switch Initialization |
| Level\_t getSwitchStatus(Switch\_t Switch\_ID) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: Switch\_ID: Switch ID  **Return**: Switch Level; High/Low  **Description:** Switch Get Status |
| Door Sensor | Status\_t initDoorSensor(Sensor\_t Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: Sensor Config  **Return**: Succeed/Fail Status  **Description**: Sensor Initialization |
| Void getDoorData(uint8 \* Buffer) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Buffer**: Buffer for Read Data  **Return**: void  **Description**: Read Sensor Data |
| Speed Sensor | Status\_t initSpeedSensor(Sensor\_t Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: Sensor Config  **Return**: Succeed/Fail Status  **Description**: Sensor Initialization |
| Void getSpeedData(uint8 \* Buffer) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Buffer**: Buffer for Read Data  **Return**: void  **Description**: Read Sensor Data |
| Comm Handler | Status\_t initCommHandler(Comm\_t Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: Communication Config  **Return**: Succeed/Fail Status  **Description**: Comm Initialization |
| Status\_t commReceiveData(uint64 \* Buffer) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Buffer**: Buffer for Data  **Return**: Succeed/Fail Status  **Description**: Comm Receive Data |
| Status\_t comTransmitData(uint64 Data) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Data**: Data to be sent  **Return**: Succeed/Fail Status  **Description**: Comm Transmit Data |
| Layer | Module | API | **Description** |
| MCAL | GPIO | Status\_t initGPIO(GPIO\_Cfg Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: GPIO Config Structure  **Return**: Success/Fail Status  **Description:** GPIO Initialization |
| Level\_t readGPIO(Port\_t port, Pin\_t pin) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Port**: Port to be accessed  **Pin**: Pin to be accessed  **Return**: Pin Level; High/Low  **Description:** Read Pin Value |
| void writeGPIO(Port\_t port, Pin\_t pin, Level\_t Level) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Port**: Port to be accessed  **Pin**: Pin to be accessed  Level: Pin Level; High/Low  **Return**: void  **Description:** Write Pin Value |
| ADC | Status\_t initADCChannel(ADC\_Cfg Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: ADC Config  **Return**: Succeed/Fail Status  **Description**: ADC Initialization |
| Void getADCValue(uint8 \* Buffer) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Buffer**: Buffer for Read Data  **Return**: void  **Description**: Read ADC Data |
| Timer | Status\_t initTimer(Timer\_Cfg Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: Timer Config  **Return**: Succeed/Fail Status  **Description**: Timer Initialization |
| Void startTimer(void) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Buffer**: void  **Return**: void  **Description**: Start Timer Count |
| Void stopTimer(void) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Buffer**: void  **Return**: void  **Description**: Stop Timer Count |
| CAN | Status\_t initCANBus(CAN\_Cfg Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: CAN Config  **Return**: Succeed/Fail Status  **Description**: CAN Initialization |
| Status\_t canReceiveData(uint64 \* Buffer) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Buffer**: Buffer for Data  **Return**: Succeed/Fail Status  **Description**: CAN Receive Data |
| Status\_t canTransmitData(uint64 Data) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Data**: Data to be sent  **Return**: Succeed/Fail Status  **Description**: CAN Transmit Data |

**APIs Types Definitions**

|  |  |
| --- | --- |
| **Type** | **Definition** |
| Uint8 | typedef unsigned char |
| Uint64 | typedef unsigned long long |
| Status\_t | typedef enum{FAIL, SUCCESS} |
| Level\_t | typedef enum{LOW, HIGH} |
| Sensor\_t | typedef enum{DOOR\_SENSOR, SPEED\_SENSOR} |
| Switch\_t | typedef enum{SWITCH\_0, SWITCH\_1} |
| Port\_t | typedef enum{PORT\_0, PORT\_1, …} |
| Pin\_t | typedef enum{PIN\_0, PIN\_1, …} |
| Switch\_cfg | Configuration for Switch |
| GPIO\_cfg | GPIO Configurations |
| Sensor\_cfg | Sensor Configurations |
| ADC\_cfg | ADC Configurations |
| Timer\_cfg | Timer Configuration |
| CManager\_cfg | Communication Manager Configurations |
| Comm\_cfg | Communication Handler Configurations |
| CAN\_cfg | CAN Bus Configurations |

1. **Folder Structure**

|  |  |  |
| --- | --- | --- |
| **Folder Name** | **C Files** | **H Files** |
| Application | main |  |
| Service | CommManager  CommManager\_cfg  SensorManager  SensorManager\_cfg | CommManager  CommManager\_cfg  SensorManager  SensorManager\_cfg |
| On-Board Layer | CommHandler  CommHandler\_cfg  DoorSensor  DoorSensor\_cfg  SpeedSensor  SpeedSensor\_cfg  LighrSwitch  LightSwitch\_cfg | CommHandler  CommHandler\_cfg  DoorSensor  DoorSensor\_cfg  SpeedSensor  SpeedSensor\_cfg  LighrSwitch  LightSwitch\_cfg |
| MCAL | GPIO  GPIO\_cfg  ADC  ADC\_cfg  CAN  CAN\_cfg  Timer  Timer\_cfg | GPIO  GPIO\_cfg  ADC  ADC\_cfg  CAN  CAN\_cfg  Timer  Timer\_cfg |
| OS | OS | OS |
| Common |  | Std\_types  Common\_macros |

**ECU 2**

1. **Layered Architecture**

Chart, bar chart

Description automatically generated

1. **ECU Components & Modules**

#### ECU Interface:

1. CAN Communication
2. Buzzer
3. Left Light
4. Right Light

#### Interface Modules:

1. GPIO Module
2. GPT Module
3. CAN Module
4. **Modules APIs**

MCAL & Service Layer have the same modules needed in MCU 2 App

|  |  |  |  |
| --- | --- | --- | --- |
| **Layer** | **Module** | **API** | **Description** |
| Application Layer | Main Application | Void BuzzerTask(void) | **Sync/Async:** Synchronous  **Reentrancy**: Non-Reentrant  **Args**: Void  **Return**: Void  **Description:** Control Buzzer |
| Void LiftLightTask(void) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: Void  **Return**: Void  **Description**: Control Left Light |
| Void RightLightTask(void) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: Void  **Return**: Void  **Description**: Control Right Light |
| Void ReceiveDataTask(void) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: Void  **Return**: Void  **Description**: Receive MCU1 Data |

|  |  |  |  |
| --- | --- | --- | --- |
| Layer | Module | API | **Description** |
| On-Board Layer | Buzzer | Status\_t initBuzzer(Buzzer\_Cfg Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: Buzzer Config Structure  **Return**: Success/Fail Status  **Description:** Buzzer Initialization |
| Status\_t buzzerOn(Buzzer\_t Buzzer\_ID) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: Buzzer\_ID: Buzzer ID  **Return:** void  **Description:** Turn on Buzzer |
| Status\_t buzzerOff(Buzzer\_t Buzzer\_ID) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: Buzzer\_ID: Buzzer ID  **Return:** void  **Description:** Turn off Buzzer |
| Left Light | Status\_t initLeftLight(Light\_t Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: Light Config  **Return**: Succeed/Fail Status  **Description**: Light Initialization |
| Status\_t leftLightOn(Light\_t Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: Light Config Structure  **Return**: Success/Fail Status  **Description**: Turn on Light |
| Status\_t leftLightOff(Light\_t Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: Light Config Structure  **Return**: Success/Fail Status  **Description**: Turn off Light |
| Right Light | Status\_t initRightLight(Light\_t Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: Light Config  **Return**: Succeed/Fail Status  **Description**: Light Initialization |
| Status\_t RightLightOn(Light\_t Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: Light Config Structure  **Return**: Success/Fail Status  **Description**: Turn on Light |
| Status\_t RightLightOff(Light\_t Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: Light Config Structure  **Return**: Success/Fail Status  **Description**: Turn off Light |
| Comm Handler | Status\_t initCommHandler(Comm\_t Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: Communication Config  **Return**: Succeed/Fail Status  **Description**: Comm Initialization |
| Status\_t commReceiveData(uint64 \* Buffer) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Buffer**: Buffer for Data  **Return**: Succeed/Fail Status  **Description**: Comm Receive Data |
| Status\_t comTransmitData(uint64 Data) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Data**: Data to be sent  **Return**: Succeed/Fail Status  **Description**: Comm Transmit Data |
| Layer | Module | API | **Description** |
| MCAL | GPIO | Status\_t initGPIO(GPIO\_Cfg Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: GPIO Config Structure  **Return**: Success/Fail Status  **Description:** GPIO Initialization |
| Level\_t readGPIO(Port\_t port, Pin\_t pin) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Port**: Port to be accessed  **Pin**: Pin to be accessed  **Return**: Pin Level; High/Low  **Description:** Read Pin Value |
| void writeGPIO(Port\_t port, Pin\_t pin, Level\_t Level) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Port**: Port to be accessed  **Pin**: Pin to be accessed  Level: Pin Level; High/Low  **Return**: void  **Description:** Write Pin Value |
| ADC | Status\_t initADCChannel(ADC\_Cfg Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: ADC Config  **Return**: Succeed/Fail Status  **Description**: ADC Initialization |
| Void getADCValue(uint8 \* Buffer) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Buffer**: Buffer for Read Data  **Return**: void  **Description**: Read ADC Data |
| Timer | Status\_t initTimer(Timer\_Cfg Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: Timer Config  **Return**: Succeed/Fail Status  **Description**: Timer Initialization |
| Void startTimer(void) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Buffer**: void  **Return**: void  **Description**: Start Timer Count |
| Void stopTimer(void) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Buffer**: void  **Return**: void  **Description**: Stop Timer Count |
| CAN | Status\_t initCANBus(CAN\_Cfg Cfg) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Cfg**: CAN Config  **Return**: Succeed/Fail Status  **Description**: CAN Initialization |
| Status\_t canReceiveData(uint64 \* Buffer) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Buffer**: Buffer for Data  **Return**: Succeed/Fail Status  **Description**: CAN Receive Data |
| Status\_t canTransmitData(uint64 Data) | **Sync/Async**: Synchronous  **Reentrancy**: Non-Reentrant  **Args**: **Data**: Data to be sent  **Return**: Succeed/Fail Status  **Description**: CAN Transmit Data |

**APIs Types Definitions**

|  |  |
| --- | --- |
| **Type** | **Definition** |
| Uint8 | typedef unsigned char |
| Uint64 | typedef unsigned long long |
| Status\_t | typedef enum{FAIL, SUCCESS} |
| Buzzer\_t | typedef enum{BUZZER\_0, BUZZER\_1,…} |
| Light\_cfg | Configuration for Light |
| GPIO\_cfg | GPIO Configurations |
| Buzzer\_cfg | Buzzer Configurations |
| Timer\_cfg | Timer Configuration |
| CManager\_cfg | Communication Manager Configurations |
| Comm\_cfg | Communication Handler Configurations |
| CAN\_cfg | CAN Bus Configurations |

1. **Folder Structure**

|  |  |  |
| --- | --- | --- |
| **Folder Name** | **C Files** | **H Files** |
| Application | main |  |
| Service | CommManager  CommManager\_cfg | CommManager  CommManager\_cfg |
| On-Board Layer | CommHandler  CommHandler\_cfg  Buzzer  Buzzer\_cfg  RightLight  RightLight\_cfg  LeftLight  LeftLight\_cfg | CommHandler  CommHandler\_cfg  Buzzer  Buzzer\_cfg  RightLight  RightLight\_cfg  LeftLight  LeftLight\_cfg |
| MCAL | GPIO  GPIO\_cfg  CAN  CAN\_cfg  Timer  Timer\_cfg | GPIO  GPIO\_cfg  CAN  CAN\_cfg  Timer  Timer\_cfg |
| OS | OS | OS |
| Common |  | Std\_types  Common\_macros |