

<Req_PO7_SVC_GDD.doc>

Global Design Document

Version 1.0

2/3/2023

Table of Contents

Contents

1.	Introduction	4
1.1	Purpose of the GDD	4
2.	Software context diagram.....	5
3.	Input output signals.....	6
4.	Static Architecture	7
5.	Component APIs.....	8
5.1	LCD.....	8
5.2	BUZZER	9
5.3	MOTORS	9
5.4	DOOR_Sensor	9

List of Tables

Table 1 - Status	3
Table 2 – Document History	3
Table 3 - Referenced Documents.....	3

Status:

Version	Status	Author	Date
1.0	Draft	Peter Samy	1/3/2023
1.1	Proposed	Kareem Murad	2/3/2023
1.2	Done	Peter & Kareem	3/3/2023

Document history:

Version	Change	Author	Date
1.0	Initial creation	Peter Samy	1/3/2023
1.1	Updating voltage threshold to 6V instead of 6.4V in requirement Req_ID	Kareem Murad	2/3/2023
1.2	Editing the limited current	Peter & Kareem	3/3/2023

Reference documents:

Ref. number	Doc. name	version	Status
1.0	Draft	1.0	Released
1.1	Proposed	1.1	Released

1. Introduction

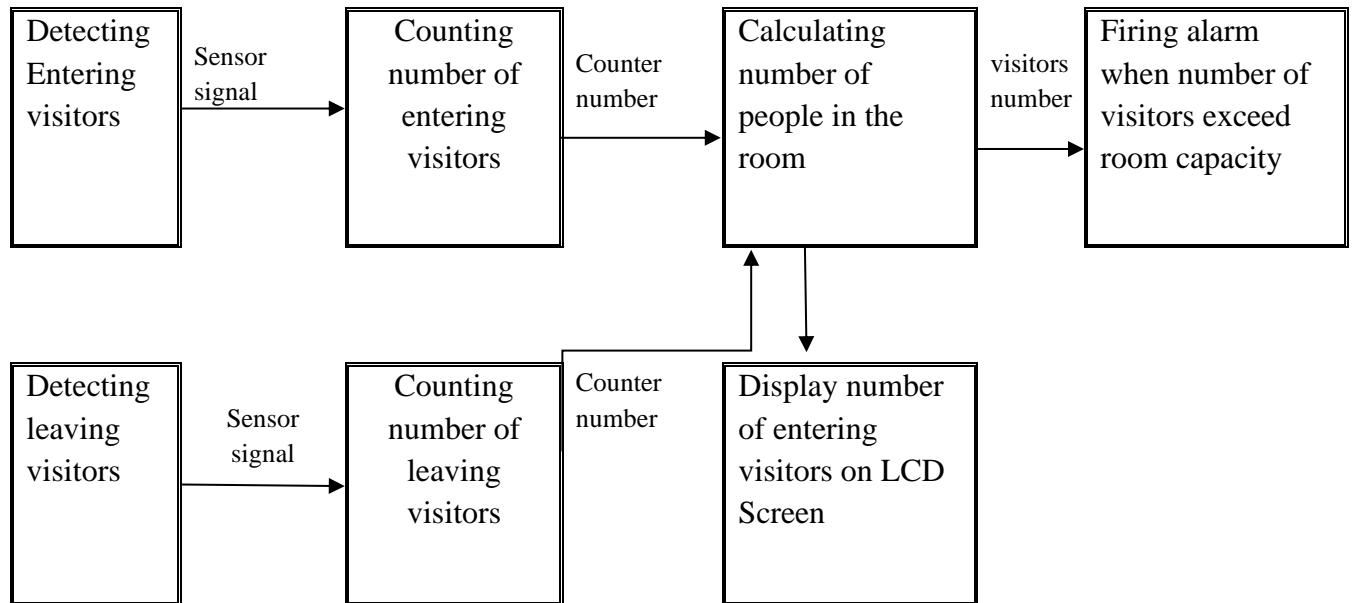
The Global Design Document (GDD) describes how the functional and nonfunctional requirements recorded in the Requirements Document, the preliminary user-oriented functional design recorded in the High Level Technical Design Concept/Alternatives document, and the preliminary data design documented in the Logical Data Model (LDM) transform into more technical system design specifications from which the system is built. The GDD documents the high-level system design and the low-level detailed design specifications.

The GDD describes design goals and considerations, provides a high-level overview of the system architecture, and describes the data design associated with the system, as well as the human-machine interface and operational scenarios. The high-level system design is further decomposed into low-level detailed design specifications for each system component, including hardware, internal communications, software, system integrity controls, and external interfaces. The high-level system design serves as primary input to the Preliminary Design Review (PDR). The low-level detailed design serves as input to the Detailed Design Review (DDR).

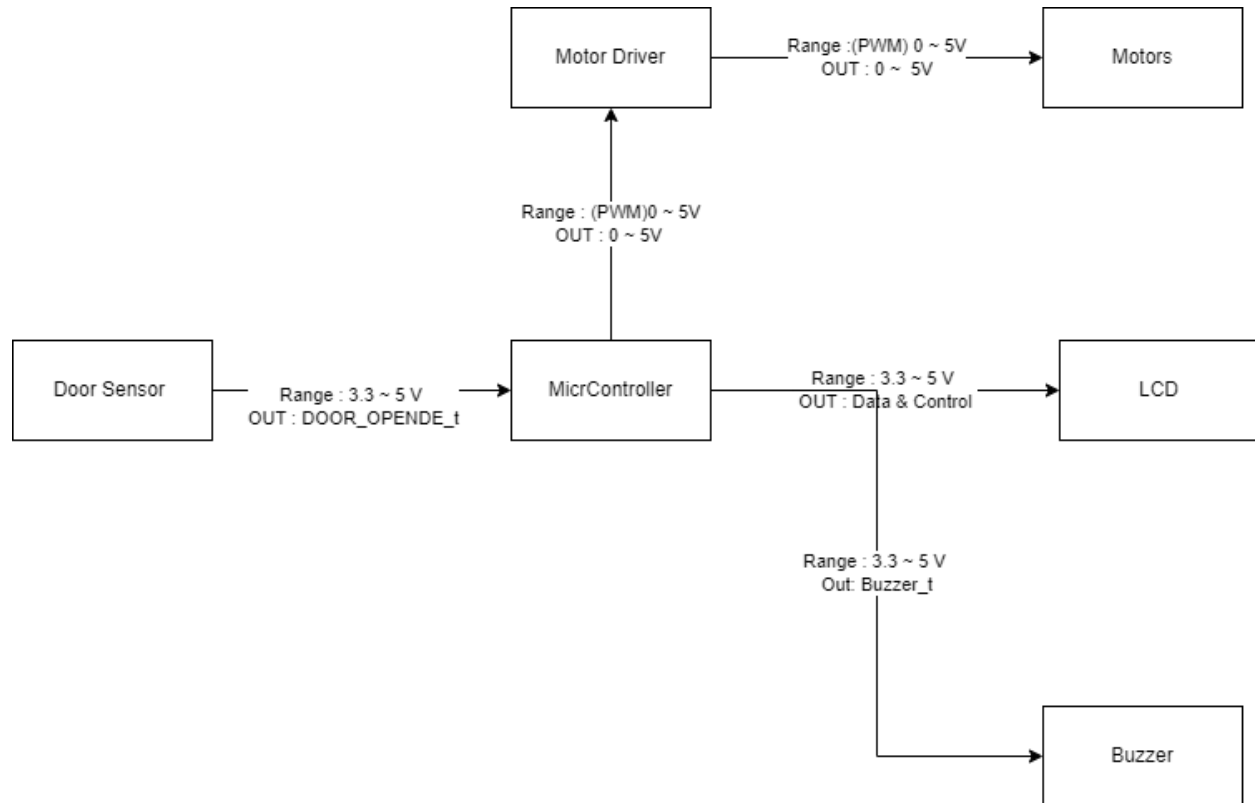
1.1 Purpose of the GDD

The GDD documents and tracks the necessary information required to effectively define architecture and system design in order to give the development team guidance on the architecture of the system to be developed. Design documents are incrementally and iteratively produced during the system development life cycle, based on the particular circumstances of the information technology (IT) project and the system development methodology used for developing the system. Its intended audience is the project manager, project team, and development team. Some portions of this document, such as the user interface (UI), may be shared with the client/user, and other stakeholders whose input/approval into the UI is needed.

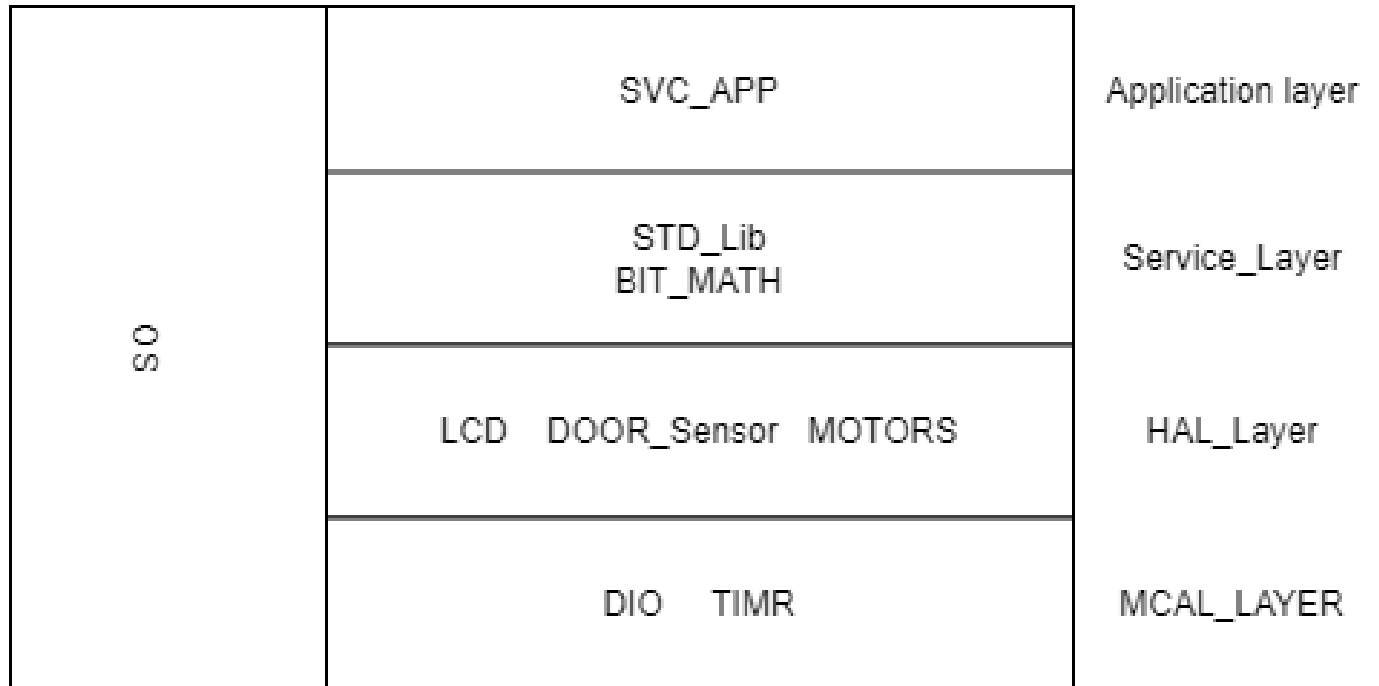
2. Software context diagram



3. Input output signals



4. Static Architecture



5. Component APIs

5.1 LCD

Return Type	API_NAME	Input arguments	Description
Void	LCD_vInit_4Bit	Void	Initialize the LCD module in 4 Bit mode.
Void	LCD_vInit_8Bit	Void	Initialize the LCD module in 8 Bit mode.
Void	LCD_Send_Command	Command_t	Sends commands to the LCD (eg: Clear screen) From the Command_t list.
Void	LCD_Send_Data	Data_t	Sends data to the LCD, from the Data_t list and prints it on the screen.
Void	LCD_Send_String	String	Sends a string to the LCD, and prints it on the screen.
void	LCD_Gotoxy	posX , posY	Send command to the LCD to move the cursor to a specific x,y coordinate.
void	LCD_Send_Int	Number	Sends a number to the LCD, and prints it on the screen.

5.2 BUZZER

Retun Type	API_NAME	Input arguments	Description
Void	Buzzer_init	Void	Initialize the BUZZER module.
Void	Buzzer_On	Void	Turns the BUZZER ON
Void	Buzzer_OFF	Void	Turns the BUZZER OFF

5.3 MOTORS

Retun Type	API_NAME	Input arguments	Description
Void	Motor_dutycycle_config	DesiredDutycycle	Initialize the MOTOR module, with the desired duty cycle.
Void	Motor_clock_wise	Void	Turn the the motor on in clock wise direction.
Void	Motor_anti_clock_wise	Void	Turn the the motor on in anti-clock wise direction.

5.4 DOOR_Sensor

Retun Type	API_NAME	Input arguments	Description
Void	Door_Sensor_init	Void	Initialize the SENSOR module.
Void	Door_Open	Void	Open the door if the sensor gives a signal.
Void	Door_Close	Void	Close the door if the sensor doesn't give a signal.