Req\_ PO4\_DGELV \_Switch\_CDD

for

Digital Elevator PO4\_DGELV

Version 1.0 Proposed

**Prepared by /   
- Ahmed Refaat**

March 5, 2020

Document Status

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Document Status** | **Author** | **Date** |
| V\_1.0 | Proposed | Ahmed Refaat | March 5, 2020 |

Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Version** | **Author** | **Date** | **Change Description** | **Document Status** |
| Req\_ PO4\_DGELV \_Switch\_CDD | V\_1.0 | Ahmed Refaat | March 5, 2020 | Initial creation of the Switch CDD Document by adding  1) Naming convention  2) Document status  3) Revision history  4) Context diagram  5) APIs | Proposed |
|  |  |  |  |  |  |

Reference Table

|  |  |  |
| --- | --- | --- |
| **Ref. Document** | **Version** | **Document Status** |
| SRS\_DIGITAL\_ELEVATOR | V\_1.4 | Proposed |
| GDD\_DIGITAL\_ELEVATOR | V\_1.2 | Proposed |
|  |  |  |

Contents

[Revision History 3](#_Toc34243918)

[Reference Table 4](#_Toc34243919)

[1. Introduction 6](#_Toc34243920)

[1.1 Purpose 6](#_Toc34243921)

[1.2 Project Scope 6](#_Toc34243922)

[2. Software Context Diagram 7](#_Toc34243923)

[DIO\_Driver 7](file:///F:\ITI%20-%20embedded%20track\Switch_CDD.docx#_Toc34243924)

[3. Switch flowchart 8](#_Toc34243925)

[4. Switch APIs 9](#_Toc34243926)

# Introduction

## 1.1 Purpose

This project aims at developing a Digital Elevator with lock system to be more secure and have specific functionalities.

The purpose of this document is to present a detailed description of the switch component in the Digital Elevator System. It will explain the software context and APIs of the component.

## 1.2 Project Scope

This software system will be an Embedded System for a digital elevator. This system will be designed to secure the usage of the elevator and handle the movement of it. By having a limited number of resigned users with unique ID and entered password, we can secure the usage of the elevator. Using some developed buttons, the users can easily control the movement of the elevator.

# Software Context Diagram

DIO\_Driver

GetSwitchState

Elevator\_control

Switch handler

Handler

DIO\_GetPinValue

# Switch APIss

|  |  |
| --- | --- |
| Req. ID | Req\_DIGELV\_CDD\_001\_V1.0 |
| Covers | Req\_DIGELV\_GDD\_004\_V1.1 |
| Component Name | Switch\_Driver |
| API Name | Error\_Status GetSwitchState(u8 Copy\_u8SwitchNum ,u8 \*Copy\_u8SwitchValue); |
| Return type | It’s u8 Error\_Status, it returns E\_OK or E\_NO   |  |  | | --- | --- | | E\_OK | 0 | | E\_NOK | 1 | |
| Parameters type | u8 Copy\_u8SwitchNum   |  |  | | --- | --- | | UP\_SWITCH | 0 | | DOWN\_SWITCH | 1 | | RESET\_SWITCH | 2 |   u8 \*Copy\_u8SwitchValue  it’s a pointer where the read value will be written in   |  |  | | --- | --- | | SW\_PRESSED | 1 | | SW\_RELEASED | 0 | |
| Description | The functionality of this API to read the switch value by receiving the read value in a pointer. |

**Algorithm Flow chart**

Start

Yes

No

If

(SwitchNum) isValid

\* SwitchValue = Local\_u8PinValue

LocError = E\_OK

LocError = E\_NOK

End