Software Design Document (SDD)

**Communication Protocol Emulation**

Version: 1.00

Date: 16-02-2021



Volansys Technologies Private Limited

Block A - 7th Floor, Safal Profitaire,

Corporate Road, Prahladnagar,

Ahmedabad, Gujarat 380015

India

[www.volansys.com](http://www.volansys.com)

Phone: 91 79 4004 1994

Email: [info@volansys.com](mailto:info@volansys.com)

**Confidentiality Notice**

Copyright © 2019 Volansys – All rights reserved.

This document is authored by Volansys and is Volansys’ intellectual property, including the copyrights in all countries in the world. This document is provided under a license to use only with all other rights, including ownership rights, being retained by Volansys. This file may not be distributed, copied, or reproduced in any manner, electronic or otherwise, without the express written consent of Volansys.

**Table of contents**

[**1**](#_heading=h.gjdgxs) **Document Details 3**

[1.1](#_heading=h.30j0zll) Revision History 3

[1.2](#_heading=h.1fob9te) References 3

[1.3](#_heading=h.3znysh7) Definitions 3

[1.4](#_heading=h.2et92p0) Acronyms & Abbreviations 3

[**2**](#_heading=h.tyjcwt) **System Overview 4**

[2.1](#_heading=h.3dy6vkm) Theory of Operation 4

[**3**](#_heading=h.2s8eyo1) **System Architecture 5**

[3.1](#_heading=h.17dp8vu) Interfaces 5

[3.2](#_heading=h.3rdcrjn) Architecture 6

[**4**](#_heading=h.2jxsxqh) **Detailed Design 8**

[4.1](#_heading=h.z337ya) Data Design 8

[4.3](#_heading=h.1y810tw) User Interface Design **11**

[**7**](#_heading=h.2bn6wsx) **Appendix** 11

**Table of Figures**

Figure 1: System diagram 4

Figure 2: Flow of project 5

Figure 3: Validate\_IP\_Frame 6

Figure 4: Store\_Single\_TLV\_Frame 6

Figure 5: Store\_Group\_TLV\_Frame 7

# Document Details

## Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version#** | **Details of change** | **Author** | **Reviewers** |
| E.g. | E.g. |  |  |  |
| 1-Sep-2013 | Draft 0.1 |  |  |  |

## References

|  |  |
| --- | --- |
| **Document/Link** | **Remark** |
| Refer requirement specs here |  |
|  |  |

## Definitions

|  |  |
| --- | --- |
| **Term** | **Definition** |
|  |  |
|  |  |

## Acronyms & Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| FW | Firmware |
| SW | Software |

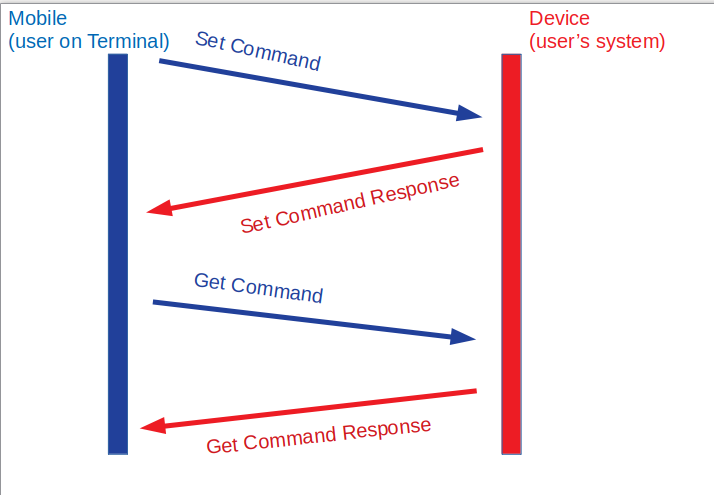
# System Overview

This project will simulate a scenario in which the set and get commands are sent from a mobile (User input on terminal in our case) and get appropriate response from the device (response will be generated from the users system in our case) in BLE command frame (TLV) format. In short this product will be a TLV frame emulator.

## Theory of Operation

Users will need to manually enter the BLE command frame (TLV) on the terminal acting as a mobile device. Device response will be simulated by the users system only.

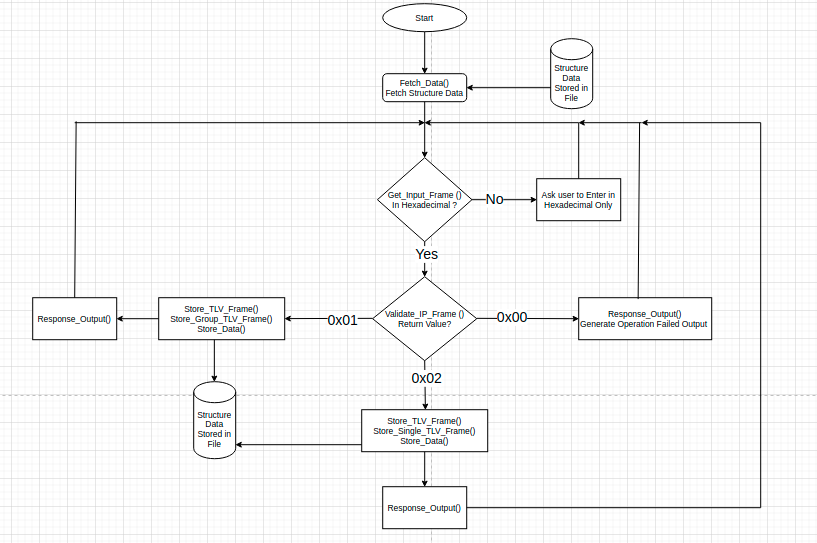
A TLV frame from the terminal will be sent to the system for set or get command then an appropriate response will be generated in the TLV structure itself in the users system and will be sent as a response to the user on the terminal.



**Figure 1: System diagram**

# System Architecture

This section is to describe top level architecture of the system. This is similar to high level design.

**

**Figure 2: Flow of Project**

**Fetch\_Data()**

It will fetch the data of structures from the file.

**Get\_Input\_Frame()**

It will take input as a character array then convert it to hex array for further processing also handles the inappropriate input for the program.

**Validate\_IP\_Frame()**

This will validate the template of the input array received if it’s correct then it will process it further for range validation

**Response\_Output()**

generate appropriate output for the user as per set and get command.

**Store\_TLV\_Frame()**

Stores the received input array in TLV structure.

**Store\_Single\_TLV\_Frame()**

Stores the value in frame structure for single command

**Store\_Group\_TLV\_Frame()**

Stores the value in frame structure for group command

## Interfaces

All the actual communication happens over BLE between a BLE Device and a smartphone.

## Architecture

# 

**Figure 3: Validate IP Frame Function**

**Validate\_Group\_IP\_Frame()**

Verifies the input with the template of Group Frame.

**Validate\_Single\_IP\_Frame()**

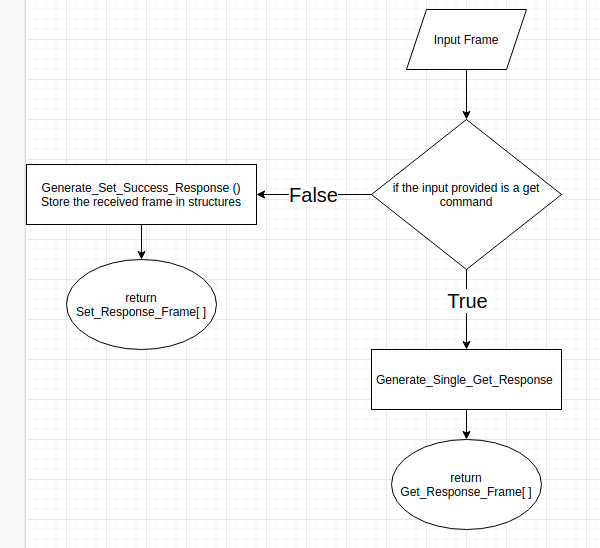
Verifies the input with the template of Single Frame.

**Range\_Validate()**

Verifies the range as per the command id in the input frame.

**Genrerate\_Set\_Failed\_Response()**

Generate failed response for the set command as range requirements are not matched.



**Figure 4: Store Single TLV Frame Function**

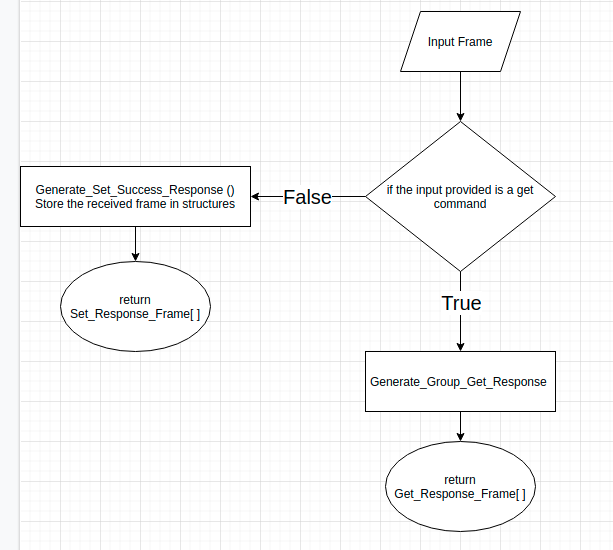
**Genrerate\_Single\_Get\_Response()**

Generate response frame for single get command.

**Generate\_Set\_Success\_Response()**

Generate Set command success frame.

Set\_Response\_Frame and Get\_Response frame are generated in this module as per the requirement of the received frame.



**Figure 5: Store Group TLV Frame Function**

**Genrerate\_Set\_Success\_Response()**

Generate response frame for successful set command.

**Generate\_Group\_Get\_Response()**

Generate the get command response for group command.

# 3 Detailed Design

## Data Design

/\* BLE TLV Structure \*/

typedef struct

{

BYTE Type;

BYTE Length;

BYTE Value[MAX\_VALUE\_SIZE];

} BLE\_Data\_TLV;

BLE\_Data\_TLV TLV\_Frame;

/\* Structures For Storing Value \*/

/\* Advertising Frequency Configuration \*/

typedef struct

{

BYTE\_4 Normal\_Adv\_Freq;

BYTE\_4 Checkin\_Adv\_Freq;

BYTE\_4 HIE\_Adv\_Freq;

BYTE\_4 Alert\_Adv\_Freq;

} Adv\_Freq\_Config;

Adv\_Freq\_Config Adv\_Freq;

/\* Advertising Timeout Configuration \*/

typedef struct

{

BYTE\_4 Normal\_Adv\_Timeout;

BYTE\_4 Checkin\_Adv\_Timeout;

BYTE\_4 HIE\_Adv\_Timeout;

BYTE\_4 Alert\_Adv\_Timeout;

} Adv\_Timeout\_Config;

Adv\_Timeout\_Config Adv\_Timeout;

/\* Accelerometer Configuration \*/

typedef struct

{

BYTE Acc\_Scale;

BYTE\_2 Acc\_Wakeup\_Threshold;

} Acc\_Config;

Acc\_Config Acc;

/\* Impact Configuration \*/

typedef struct

{

BYTE\_2 Emfit\_Sensing\_Threshold;

BYTE\_2 Emfit\_Sample\_Rate;

BYTE\_2 Emfit\_Total\_Sample;

} Impact\_Config;

Impact\_Config Impact;

/\* Device Configuration \*/

typedef struct

{

BYTE\_4 Checkin\_Interval;

BYTE\_4 Inactive\_Timeout;

} Device\_Config;

Device\_Config Device;

/\* HIE Configuration \*/

typedef struct

{

BYTE HIE\_Queue\_Size;

BYTE HIE\_Threshold;

BYTE\_4 HIE\_Timeout;

} HIE\_Config;

HIE\_Config HIE;

## User Interface Design

Simple printf() will be used to display instructions and output on terminal. scanf() will be used to take input from the user.

**Template Revision History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Security Classification** | | | Volansys Confidential | | |
| **Distribution** | | | Restricted | | |
| **Prepared By** | | | Engineering | | |
| **Maintained By** | | | SEPG | | |
| **Approved By** | | | Engineering Head | | |
| **Revision** | **Date** | **Details of Change** | **Author** | **Reviewer** | **Approver** |
| 0.1 | 14-Nov-17 | First draft | PA |  |  |
| 0.2 | 31-Dec-17 | Updated formatting and review track changes | PA |  |  |
| 1.0 | 01-Feb-18 | Baseline by SEPG | SL | DS/RSR | PS |
| **1.1** | **31-Jul-19** | 1. **Company Logo has been updated** 2. **Spelling Error Corrected in Address & Baselined** | **Sandhya** | **Ravi** | **Dhaval** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |