

Security Classification: **Confidential**

FPT SOFTWARE ACADEMY

NguyenTC8Mock Project

# S-Record - Full Phase Design

**Version – 1.0**

**15th Jan 2024**

**Offshore Software Development**

# Table of Contents

[Table of Contents 2](#_TOC_250005)

1. [INTRODUCTION 5](#_TOC_250004)
   1. [Scope 5](#_TOC_250003)
   2. [Definitions, Acronyms and Abbreviations 5](#_TOC_250002)
   3. [Overview 5](#_TOC_250001)
   4. [References 6](#_TOC_250000)
2. SOFTWARE REQUIREMENTS 7
3. DESIGN 8
   1. System Overview 8
   2. Sequence diagram 9
      1. Feature 1 9
      2. Feature 2 11
   3. Class diagram 13
      1. Feature 1 13
      2. Feature 2 13
4. IMPLEMENTATION 14
   1. VendorExt-Project XXXX 14
   2. Source code structure 14
      1. Source code file name of components: 14
      2. Directory structure 14
   3. Common 15
      1. Constants 15
      2. Structure list 15
      3. Function list 16
   4. Features Detail 18
      1. Feature 1 18
      2. Feature 2 19
5. UNIT TEST 22
   1. Unit Test plan 22
      1. Prerequisite 22
      2. Testing environment 22
      3. Convention 22
      4. Assumptions 22
   2. Unit Test cases 23
      1. Common Functions Test cases 23
      2. Features Test cases 23
   3. Result of the Unit Test. 23
6. OPEN ISSUES 23
7. APPENDIX 23

**RECORD OF CHANGES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Version History** | | | |
| **Version** | **Date** | **Author** | **Details** |
| 0.1 | 15th Jan, 2024 | Nguyễn Minh Phú | New Creation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**APPROVERS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Distribution List** | | | | |
| **Name** | **Role** | **Signature** | **Date** | **Version** |
| NguyenTC8 | Trainer |  |  | 2.0 |

**CONFIDENTIALITY**

N/A

## INTRODUCTION

The document is to describe how Mock Project’ components work in low level view.

### Scope

N/A

### Definitions, Acronyms and Abbreviations

**Table 1: Abbreviations in this document**

|  |  |
| --- | --- |
| **Acronym** | **Reference** |
| ACS | Automatic Configuration Server |
| IP | Internet Protocol |
| VSCode | Visual Studio Code |
|  |  |

### Overview

This document includes following sections:

##### Introduction:

This section introduces some overview information about Mock Project.

##### Software Requirement:

This section contains excerpted parts from the requirement (SRS document) which relate to Mock Project to focus the functionality.

##### Design:

This section includes diagrams which describe functionality, from outline to detail, in

Mock Project. The diagrams are block diagram, sequence diagram, and others.

##### Implementation:

This section describes followings:

* + - * List of parameters which are corresponding to requirement document.
      * Source code’s directory structure of Mock Project.
      * Description about template files’ source code.
      * List of constants, enum types, structures, other user-defined data types which are used.
      * List of functions with detailed descriptions for using in source code.
      * Some notice when coding Mock Project.

##### Unit test:

N/A

### References

|  |  |
| --- | --- |
| **Documentation Name** | **Description** |
| HCM23\_FRF\_EMB\_09 - C Module - Mock Project.pdf | System Requirement Specification document which contains requirement of all features. |
| https://en.wikipedia.org/wiki/SREC\_(file\_format) | Technical specification about S-Record format. |
|  |  |

**2**

## SOFTWARE REQUIREMENTS

**Table 2: Software requirement table for Mock Project**

|  |  |  |
| --- | --- | --- |
| **Module YYYY** | | |
| **No.** | **Requirement** | **Note** |
| **1** | Form a sequence of integer values |  |
| **1.1** | Input from keyboard by user, to S-Records then write  them to a new S-Record file |  |
| **1.2** | Integer values are input by user are in decimal base and separated by Space character between  them |  |
| **2** | Convert 1-1 from a normal binary file to an S-Record file. The binary file contains only data, so  the application will build S-Records whose Data field is data got from a portion in binary file. |  |
| **3** | Convert binary data from a normal binary file to S-Records and add them to an existing S-Record file. (This feature is similar to the above feature, but it doesn’t create new S-Record file,  it appends S-Records to an existing file). |  |
| **4** | Check corruption for an S-Record file. |  |
| **5** | The application must have appropriate menu so that user can use conveniently |  |

## DESIGN

This section includes diagrams which describe behavior of components in Mock Project. The diagrams shown following are written in UML language using StarUML tool.

### System Overview

The following diagram shows system architecture of Mock Project

A diagram of a computer

Description automatically generated

##### Hình 1: Sơ đồ khối hệ thống Mock Project

Hiện tại, chương trình có 2 chức năng chính: Kiểm tra file S-Record, Nhập dữ liệu, Thoát chương trinh. Trong đó Nhập dữ liệu có 4 chức năng nhỏ: Ghi vào file sẵn có, ghi vào file mới, Trở về menu lựa chọn và thoát chương trình.

### Sequence diagram

### A diagram of a person with arrows Description automatically generated

##### Hình 2: Sơ đồ trình tự lựa chọn của người dùn

* + 1. **Nhập dữ liệu**

#### A diagram of a project Description automatically generatedGhi vào file mới

##### Hình 3: Sơ đồ trình tự chuyển dữ liệu sang file S-Record mới

#### Ghi vào file sẵn có

#### A diagram of a project Description automatically generated

##### Hình 4: Sơ đồ trình tự chuyển dữ liệu sang file S-Record sẵn có

### Kiểm tra file S-Record

### A diagram of a diagram Description automatically generated

##### Hình 5: Sơ đồ trình tự kiểm tra file S-Record

### Class diagram

N/A.

## IMPLEMENTATION

Git Repository : git clone git@bitbucket.org:fa-emb-cf5/nguyenminhphu.git

### VendorExt-Project XXXX

N/A

### Source code structure

* + 1. **Source code file name of components:**

##### Common:

##### Mock Project.c

### Directory structure

N/A

* 1. **Common**
     1. **Constants**

N/A

* + 1. **Structure list**

N/A

* + 1. **Function list**

|  |  |  |
| --- | --- | --- |
| int ConvertStringtoHexa (char cData[2))  {  char cHexa  ) | | |
| **Arguments** | cChar cData[2]  unsigned char cHexa | Truyền vào mảng gồm 2 phần tử  Chứa giá trị 1 byte |
| **Description:**   * + - * + Hàm dùng để chuyển đổi mảng gồm 2 kí tự sang giá trị 1 byte         + Trả về giá trị cHexa | | |
| **Flowchart:** | | |

|  |  |  |
| --- | --- | --- |
| int ConvertStringtoAddressHexa (char cData[8], unsigned char cByte)  {  char cHexa  ) | | |
| **Arguments** | Char cData[8]  unsigned char cByte  char cHexa | Truyền vào mảng gồm 8 phần tử  Truyền vào số byte  Chứa giá trị đã chuyển đổi |
| **Description:**  Hàm dùng để chuyển đổi các kí tự sang số hex theo byte truyền vào và trả về giá trị theo số byte | | |
| **Flowchart:** | | |

|  |  |  |
| --- | --- | --- |
| int GetHex (FILE \*pGet, unsigned char cNumberofbyte)  {  char cBuffer[2]  short iChecksumfunct  ) | | |
| **Arguments** | FILE \*pGet  unsigned char cNumberofbyte  char cBuffer[2]  short iChecksumfunct | Trỏ tới file đã được mở  Truyền vào số byte  Mảng chứa 2 kí tự  Giá trị trính checksum |
| **Description:**  Hàm dùng để lấy các cặp kí tự từ file chuyển đổi qua số Hex rồi cộng chúng lại với nhau  Trả về giá trị iChecksumfunct | | |
| **Flowchart:** | | |

|  |  |  |
| --- | --- | --- |
| int ConvertoNewFile (int \*pData, unsigned short iRange)  {  ) | | |
| **Arguments** | int \*pDate  unsigned short iRange | Trỏ tới dữ liệu đã được nhập  Truyền vào số lượng dữ liệu đã nhâp |
| **Description:**  Hàm dùng để ghi các giá trị người dùng nhập từ bàn phím sang file S-Record mới | | |
| **Flowchart:** | | |

|  |  |  |
| --- | --- | --- |
| int ConvertoExistingFile (int \*pData, unsigned short iRange)  {  ) | | |
| **Arguments** | int \*pDate  unsigned short iRange | Trỏ tới dữ liệu đã được nhập  Truyền vào số lượng dữ liệu đã nhâp |
| **Description:**  Hàm dùng để ghi các giá trị người dùng nhập từ bàn phím sang file S-Record sẵn có | | |
| **Flowchart:** | | |

|  |  |  |
| --- | --- | --- |
| Int Choose1 (int \*pArray, unsigned short iCount)  {  ) | | |
| **Arguments** | int \*pArray  unsigned short iCount | Trỏ tới vùng nhớ đã được cấp phát động  Truyền vào số lượng dữ liệu đã nhâp |
| **Description:**  Hàm dùng để thực hiện chức năng 1 trong menu lựa chọn | | |
| **Flowchart:** | | |

|  |  |  |
| --- | --- | --- |
| int CheckCorruption (FILE \*pCheckFunct)  {  ) | | |
| **Arguments** | FILE \*pCheckFunction | Trỏ tới file đã mở |
| **Description:**  Hàm dùng để kiểm tra xem các S-Record có hợp lệ hay không rồi in ra màn hình | | |
| **Flowchart:** | | |

|  |  |  |
| --- | --- | --- |
| int main()  {  ) | | |
| **Arguments** |  |  |
| **Description:**  Hàm chính để tùy chọn menu và gọi các hàm khác thực hiện | | |
| **Flowchart:** | | |

* 1. **Detail**
     1. **Feature 1 : Nhập dữ liệu**

#### Notes

Feature 1’ main functionality:

* + - * + Functionality 1: Ghi vào file mới
        + Functionality 2: Ghi vào file sẵn có

#### Structure List

#### N/A

* + - 1. ***Function list***

### Feature 2 : Kiểm tra file S-Record

#### Notes

N/A

* + - 1. ***Structure List***

N/A

* + - 1. ***Function list***

## UNIT TEST

### Unit Test plan

* + 1. **Prerequisite**

N/A

### Testing environment

N/A

### Convention

#### Test file name

N/A

#### Function to test

TEST(UTSuite,TestXXXX\_ID\_xx)

* + - * + xx is the ID of the test case starting from ‘01’
        + For testing static function, a wrapper function need to be placed in where can called to static function.
        + Wrapper function name : STUB\_{name of static function to be tested}

### Assumptions

* XXXX generated source code from template will not be defined unit test
* SUCCESS = 0; FAILURE = 1
  1. **Unit Test cases**
     1. **Common Functions Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Test Name** | **Function Name** | **Description** |
| 1 |  |  |  |
| 2 |  |  |  |

* + 1. **Features Test cases**
       1. ***Feature 1***

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Test Name** | **Function Name** | **Description** |
| 3 |  |  |  |
| 4 |  |  |  |

* + - 1. ***Feature 2***

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Test Name** | **Function Name** | **Description** |
| 1 |  |  |  |
| 2 |  |  |  |

### Result of the Unit Test.

The result of Unit Test is reported in the document named as “UT Coverage Report”. The document is introduced separately.

## OPEN ISSUES

N/A

1. **APPENDIX**