Installation ManualInstallation Manual

UNI92K-4DMUNI92K-4DMW

|  |
| --- |
| ***Pattern Viewer*** |
| ***Software Requirements Specification*** |
| Rev. 1.00 |



Copyright ©UNITEST Corporation

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ver | Date | Description | Revised By | Approval |
| 1.0 | 2020.08.20 | 초안 작성 | hwpark |  |
| 1.1 | 2020.08.24 | 일정계획 추가 | hwpark |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Contents**

[1. Introduction 3](#_Toc49158368)

[1.1. Purpose 3](#_Toc49158369)

[1.2. Definitions, Acronyms, and Abbreviations 3](#_Toc49158370)

[1.3. Reference 4](#_Toc49158371)

[1.4. Overview 4](#_Toc49158372)

[2. Overall Description 4](#_Toc49158373)

[2.1. Product Perspective 4](#_Toc49158374)

[2.1.1. System Interfaces 4](#_Toc49158375)

[2.1.2. User Interfaces 5](#_Toc49158376)

[2.1.3. Software Interfaces 6](#_Toc49158377)

[2.1.4. Operations 6](#_Toc49158378)

[2.2. User Characteristics 6](#_Toc49158379)

[2.3. Constraints 6](#_Toc49158380)

[3. Unified Modeling Language 7](#_Toc49158381)

[3.1. Use Case Diagram 7](#_Toc49158382)

[3.1.1. Flow Chart 8](#_Toc49158383)

[3.2. Class Diagram 9](#_Toc49158384)

[4. Management Structure & Planning 9](#_Toc49158385)

[4.1. Task management 9](#_Toc49158386)

[4.2. Schedule planning 10](#_Toc49158387)

# Introduction

본 문서는 FPGA 에서 사용되는 PAT 파일 컴파일러에 대한 Pattern Viewer Program의 소프트웨어 요구사항(Software Requirement Specification)에 대하여 기술한다.

## Purpose

Pattern Viewer Program의 소프트웨어 요구사항을 분석하고 정의한다. 개발 방향 결정, Software Architecture 설계, Software Function Block 문서의 기준으로 사용하기 위해 작성한다.

본 문서의 기술은 소프트웨어 관점에서 기술한다. 사용자의 요구사항 사항을 분석하고 작성한 문서이다. 사용자 기준에서의 흐름도, UML을 반영한다. 따라서 사용자와 관련 기능을 공유하며 Software 요구사항(기능)이 충족하는지 살펴보는 것을 문서의 기준으로 한다. 결국 사용자의 요구사항 및 요구 기능을 누락 없이 설계에 반영하고 그러한 요구사항을 구현하기 위해서 필요한 부분을 기술하는 문서이다.

## Definitions, Acronyms, and Abbreviations

ALPG : Algorithmic Pattern Generator

UML : Unified Modeling Language

GUI : Graphical User Interface

CTRL : Controller

ARR : Array

SRS : SW Requirements Statement

SWA : Software Architecture

SWI : Software Implementation

SWT : Software Test

## Reference

* mpat file format.xlsx
* comp\_eval\_te\_op\_reg\_200726.ASC
* comp\_te\_eval\_op\_reg\_200726.PAT
* UNI610-RegFor-List\_Rev0.67\_20200804.xlsx
* UNI610-RegFor-system-control-pg\_rev0.7\_20200625.xlsx

## Overview

본 문서의 기술 범위는 소프트웨어 Module 단위이다. 구성의 기술 방향은 아래와 같다.

* 사용자가 제시하는 요구사항에 의해 요구되는 기능을 기술한다.
* 올바른 프로그램 작동를 위하여 데이터를 처리하거나 충족해야하는 제약사항 혹은 서비스를 기술한다.
* 사용자가 요구하지 않았더라도 더욱 편리하게 사용할 수 있을 기능 사항들을 기술한다.

# Overall Description

## Product Perspective

### System Interfaces

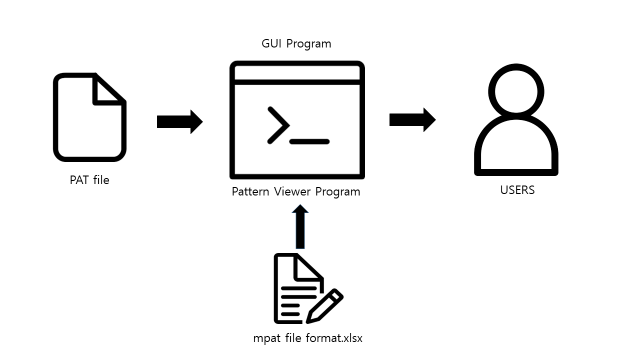


그림 1. Pattern Viewer Program System

* Pattern Viewer Program은 PAT file을 읽어들여 내용을 구성한다.
* mpat file format의 내용을 기반으로 사용자에게 정보를 전달한다.

* 사용자에게 exe 실행 파일 형식으로 전달한다.

.

### User Interfaces

* 사용자에게 컴파일된 PAT파일 내용을 GUI를 통해 제공한다.
* mpat file format을 제공하며, 기능으로는 검색 기능을 제공한다.
* 포맷 양식에서의 데이터와 일반 바이너리 양식의 데이터 두가지를 제공한다.

### Software Interfaces

* 개발 환경
  + OS : CentOS7
  + Kernel : Linux 3.10.0-1127.el7.x86\_64
  + GUI : Qt version 5.8.0
  + Editor
    - QtCreator: 4.2.1
    - VIM : Vi IMproved 7.4

### Operations

* 사용자는 exe를 실행시킨 후 PAT 파일을 선택하며 프로그램을 시작한다.
* 다른 PAT 파일을 선택하고 싶은 경우 reopen 버튼을 클릭하여 PAT 파일을 선택할 수 있다.
* 좌측은 일반적인 Binary Viewer 형태, 우측은 파일 포맷과 함께 Binary Data가 제공된다.
* 뷰 하단에 text edit에 검색하고 싶은 문자열을 입력하고 search를 클릭하여 검색한 문자열을 데이터에서 확인한다.
* 검색을 하며 검색된 문자열이 다수일 경우 next 버튼을 이용하여 커서를 다음 지점 문자열로 넘길 수 있다.
* 포맷의 오른쪽 검색창에서는 Division, Item 이름으로도 검색이 가능하다.

## User Characteristics

사용자는 ALPG에 대해서 전반적인 정보를 알고 있는 사람이어야 한다.

PAT컴파일러 동작 절차에 대해서 이해하고 있어야 한다. 뿐만 아니라 ASC파일과 컴파일된 바이너리 파일을 직접 비교하여 컴파일러가 정상동작하는지 확인할 수 있어야 한다.

UNIN610 프로젝트에 참여하고 PAT 컴파일 문법에 대해 숙지하고 있는 사용자로 가정한다.

## Constraints

* Program을 작성할 때 제공받은 mpat file format 규격에 맞도록 작성한다.
* user가 파일의 내용을 수정할 수 없고 확인만 가능하다.
* 정해진 pat양식으로만 사용가능하다.

# Unified Modeling Language

사용자의 요구 사항 분석 및 확인, 시스템 설계를 위한 객체지향 모델링 언어이다.

## Use Case Diagram

Use Case Diagram은 사용자와 시스템 사이의 상호작용을 보여주는 다이어그램이다.

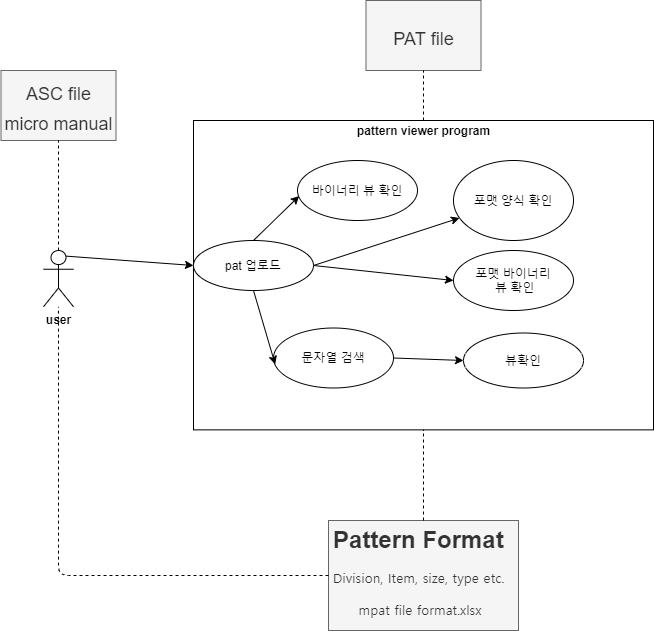


그림 2. Use Case Diagram

* 사용자는 ASC file, micro manual을 토대로 컴파일된 pat 파일이 올바른지 검증할 수 있다.
* pat파일을 업로드하고 문자열 검색, 바이너리 뷰, 포맷 바이너리 뷰, 포맷 양식 등을 확인한다.

### Flow Chart

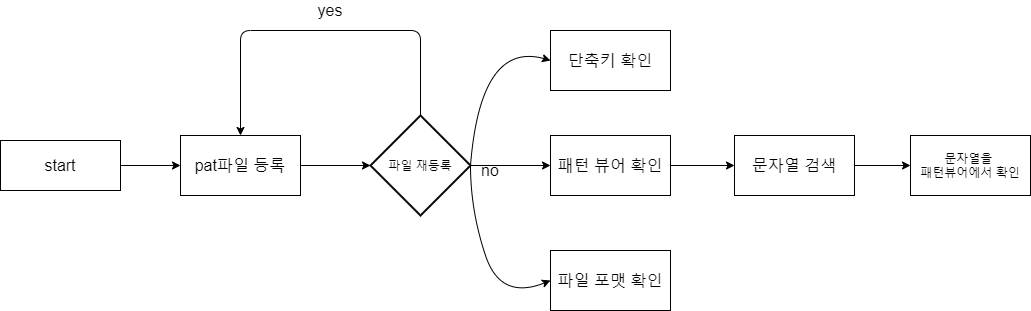


그림 3. Flow Chart

* 프로그램이 실행될 때 사용자는 pat파일을 등록한다. Reopen 버튼을 통해서 언제든지 pat 파일을 재 등록할 수 있다.
* 등록 후 변환된 바이너리 값들을 hex 값으로 확인할 수 있다.
* 특정 문자열을 입력하여 검색이 가능하다. 검색 후 기존 edit text에서 확인이 가능하다.
* 단축키와 파일 포맷도 확인이 가능하다.

## Class Diagram

Class Diagram은 클래스의 속성, 메소드, 관계를 보여주는 다이어그램이다. 그림 4와 같은 형태로 구현할 예정이다.



그림 4. Class Diagram

* widget에서 gui를 생성한다.
* file에서는 pat파일 open과 read를 담당한다. format에서는 pat file format에 맞춰 데이터를 변경한다.
* file과 format 클래스에서 데이터를 받아 widget에서 처리한다.

# Management Structure & Planning

## Task management

|  |  |  |  |
| --- | --- | --- | --- |
| **작업ID** | **작업내용** | **기간(일)** | **선행작업** |
| T1 | 요구사항 분석 및 명세(SRS) | 7 | - |
| T2 | GUI 설계 | 3 | T1 |
| T3 | 시스템 설계 (SWA) | 4 | T1, T2 |
| T4 | 구현(SWI) | 7 | T3 |
| T5 | SWT Validation | 4 | T4 |
| T6 | Manual | 3 | T1~T5 |
| T7 | 완료보고서 | 4 | T1~T5 |
| T8 | Demo | 1 | T1~T7 |
| Total | - | 33 | - |

## Schedule planning

간트 차트

나. 프로젝트 추진일정

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **구분** | **추진내용** | **추진일정 (8월 ~ 9월)** | | | | | |
| **3주** | **4주** | **1주** | **2주** | **3주** | **4주** |
| 계획 | 프로젝트의 목적 정의 및 일정산정 |  |  |  |  |  |  |
| 분석 | 프로젝트의 요구사항 분석 |  |  |  |  |  |  |
| 설계 | 인터페이스 설계 |  |  |  |  |  |  |
| 시나리오 설계 |  |  |  |  |  |  |
| 기능 설계 |  |  |  |  |  |  |
| 개발 환경 구축 |  |  |  |  |  |  |
| 시연 세트 설계 |  |  |  |  |  |  |
| 개발 | 클라이언트(Android) 어플 개발 |  |  |  |  |  |  |
| 서버(Node.js) 개발 |  |  |  |  |  |  |
| 아두이노 개발 |  |  |  |  |  |  |
| 시연 세트 제작 |  |  |  |  |  |  |
| 테스트 | 기능 테스트 |  |  |  |  |  |  |
| 연동 테스트 |  |  |  |  |  |  |
| 통합 테스트 |  |  |  |  |  |  |
| 종료 | 보고서 작성 및 종료 |  |  |  |  |  |  |
| 오프라인  미팅계획 | 진행현황 공유/의사소통 |  |  |  |  |  |  |