Application Configuration GUI for RTEMS
Software Requirements Specification

by

Group Members : **Alex Poovathingal and Jos Paul Davis**

Under Guidance of

Mrs. Bisna

Lecturer

Department of Computer Science and Engineering

Govt. Engineering College, Thrissur-9

Department of Computer Science and Engineering

Government Engineering College, Thrissur-9

April 21, 2010

Contents

1   Introduction

1.1   Purpose

The purpose of this document is to serve as a guide to students who are responsible for the engineering of the Application Configuration GUI for RTEMS project. This document is going to be verified by the mentors and potential users.

1.2   Scope

The project **Application Configuration GUI for RTEMS** aims at developing a graphical tool that can generate the RTEMS configuration directives for an application. RTEMS has to be configured for each application. Now, the configuration is done by manually adding directives in the init.c file. It's both time consuming and difficult for a non-expert user. What this project intends to do is to create a GUI tool which will give help and dialog to each and every potential configuration value accepted. It will generate the configuration directives based on the values entered by the user.

1.3   Definitions, Acronyms and Abbreviations

1.3.1   Acronyms and Abbreviations

- RTOS - Real Time Operating System

- RTEMS - Real Time Executive for Microprocessor Systems

- GUI - Graphical User Interface

- XML - Extensible Markup Language

- GTK - Gimp Tool Kit

1.3.2   Definitions

- RTOS - Real Time Operating System is an operating system intended for Real Time applications.

- XML - XML is a set of rules for encoding documents electronically.

1.4 References

RTEMS wiki http://www.rtems.org/wiki/index.php/ApplicationConfigurationGUI

RTEMS Online Documentation http://www.rtems.com/onlinedocs.html

Wikipedia wikipedia.org A free, web-based, collaborative, multilingual, encyclopedia project.

IEEE Std 830-1998(R2009) Revision of IEEE Std 830-1993 IEEE Recommended Practice for Software Requirement Specification[1]

IEEE Std 830-1984 IEEE Guide to Software Requirement Specification[2]

1.5 Overview

Rest of the document is organised as follows: Section 2 gives a general description of the software. It includes product perspective, product functions, user characteristics and assumptions and dependencies. Section 3 deals with the specific requirements the software is expected to deliver. A note on functional requirements and design constraints can be found there.

2 General Description

2.1 Product Perspective

2.1.1 System Interfaces

Even though this application is to assist RTEMS users, its not a part of RTEMS. Its an independent application which generates a file contains configuration directives which can be used for configuring RTEMS for an application.

---

[1]Sponsor: Software Engineering Standards Committee of IEEE Computer Society Approved : 25 June 1998 Reaffirmed : 9 December 2009 IEEE SA Standard Board

[2]Sponsor: Software Engineering Technical Committee of IEEE Computer Society Approved : September 30,1983 by IEEE Standards Board Approved : July 20, 1984 by American National Standards Institute

2.1.2   User Interfaces

The path containing confdefs.h and shellconf.h header files is given by the user. The application process it, creates an XML file and provides a GUI for accepting configuration data from the user. Here all configuration options RTEMS provides are displayed. Along with it, help on each of them and a dialog box which can accept values for those from the users are also present. After typing/selection the configurations, user can click on the Generate button in the GUI to generate a userconf.h header file.

2.1.3   Operations

The operations required from the user are :-

- Select the path containing the configuration header files of RTEMS. The application will generate an XML file containing the configuration options.

- Select/Insert values required for configuring RTEMS.

- Click on the save button to create an output XML file.

- Click on the generate button to create the required file.

2.2   Product Functions

There are mainly 4 functions for this applicaion :-

- From confdefs.h and shellconf.h header files, capture the configuration options available into an XML file which can be used by the application

- Using that XML file, display all the configuration options available and read their values from the user

- Create an output XML file containing the values entered by the user for each and every configuration options available

- Create an output C/H file which can be used for configuring RTEMS for an application.

DATA FLOW DIAGRAM
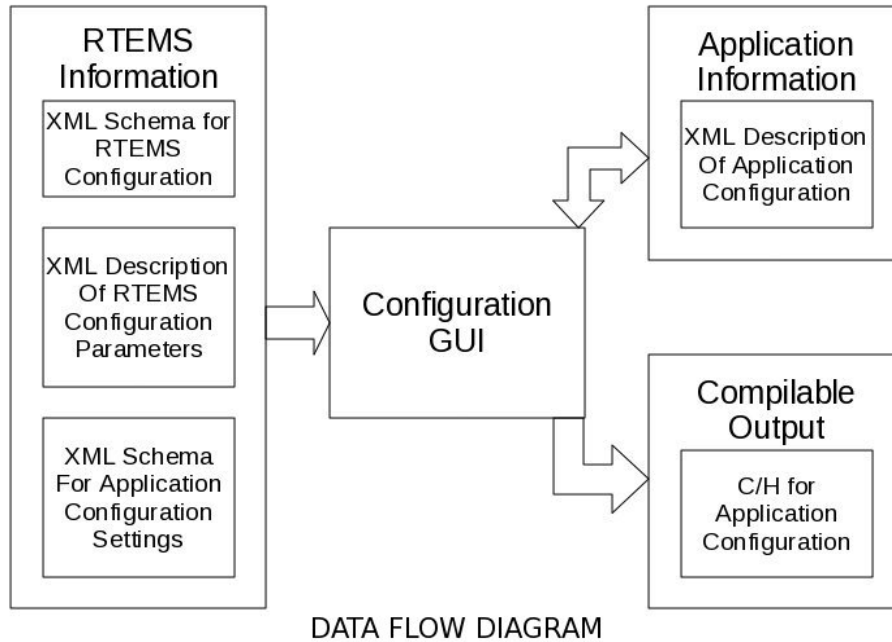
2.3 User Characteristics

The users of this application are application developers for RTEMS who wants to have a easier way to configure RTEMS than using configuration directives. The only experience expected from a user is the knowledge of configuration options available.

2.4 Assumptions and Dependencies

The application is developed using Python in a GNU/Linux system. Its expected to work in different GNU/Linux systems. But considering the portability of Python, its also expected to work with atmost slight modifications in Windows based systems too.

3 Specific Requirements

3.1 Functional Requirement

- Create a module which take the path containing confdefs.h and shell-conf.h header files and create an output XML file containing all the configuration options and its default values. This can be verified by checking the consistency of XML file and input header files

- Make a GUI application which can display the configuration options from the XML file. It should also be able to read the changes which the user gives as input. Check it for its user friendliness and ease of user.

- Create an output XML file containing the changed value. Verify its correctness with the value entered in the GUI.

- Create an C/H file from this XML file. This can be directly included in any application and used.

3.2 Design Constraints

3.2.1 Availability

To facilitate availability, the application proceeds in steps as described above. After each step an output is produced, which can be checked for correctness.

3.2.2 Maintainability

Each functions of the application can be separate modules. Hence we have modules iwth high cohesion and low coupling which makes correction and adapting easier.

3.2.3 Portability

The application is coded in Python using GTK+ which makes it highly portable. The OS specific functions used, if any, are going to be in a specific module which can be coded separately and compiled for each OS.