Table of Contents

1Introduction

1.1 Purpose of Software Configuration Management Plan

Development of software is a complicated process. Although there are a few numbers of members in the project, it is hard to manage different versions that may cause conflicts. During the process there will be a lot of individual modifications and these modifications will affect the whole project. In order to avoid the harmful effects, we have to supply a mechanism.

This mechanism is the configuration management plan and this document aims to explain the policies of this plan. This plan will help to handle the changes during the project.

1.2 The Scope of the Configuration Management Plan

The scope of this document is to describe all Configuration Management activities for our project. It describes the configuration management elements of the configuration identification, change control, status accounting, and audits and reviews. Configuration management plan enables us to identify the Software Configuration Items, manage the change control, audit the changes and report the changes to inform other team members. By this way, the concurrence of different versions is assured.

1.3 Definitions, Acronyms and Abbreviations

CI	Configuration Item
CM	Configuration Management
CSA	Configuration Status Accounting
CVS	Concurrent Versioning System
SCA	Software Configuration Audit
CMP	Configuration Management Plan
SCMP	Software Configuration Management Plan

1.4 Document References

http://www.cvshome.org

IEEE Standard (IEEE Std 828-1998) for Software Configuration Management Plan

1.5 Document Overview

In the second chapter we will describe the organization of CMP. We will give the details of organization, responsibilities, tools and infrastructure.

The third chapter is about the CM process. It will include identification, management and control, configuration status accounting, and auditing.

Finally, we will give details of our CM milestones, project resources, and plan optimization in the fourth, fifth, and sixth chapters respectively.

2The Organizations of CM Framework

2.1 Organization

In our project we need different teams for development of the modules and for testing these modules. Our interface, simulator, and upload modules development team is composed of all the members of our project; compiler module development team includes Oya and Serdar; and debugger module development team Ufuk and Beyza. For testing these modules we made teams of two: Oya and Beyza will make up the compiler, debugger and upload testing team while Ufuk and Serdar are going to work on testing the simulator. These teams will be divided into several sub-teams according to the work-packages of each module during the implementation process which will be determined by the manager of each module. In order to avoid chaos in the implementation of modules, each member of each team will participate in SCM related with that module. Usage of CVS will support the concurrency of different versions of the modules.

2.2 Responsibilities

As we stated above, our teams are responsible for different modules.

Interface module team is responsible for:

- Main window
- Simulator window
- Debugger window
- Editor

Compiler module development team is responsible for adapting the open source compiler of Microsim to our pic 16F877.

Debugger module development team is responsible for

- Showing if any error occurred during compilation
- Supporting user-defined break points
- Showing the content of registers
- Showing the current status of user-defined variables

Simulator module development team is responsible for

- Simulating leds
- Simulating seven segment displays
- Simulating LCD display
- Supporting the user inputs via switches

Test team is responsible for testing and debugging the modules individually and all together. These test processes will be done as black-box testing, and unit testing.

Upload module team will be responsible for the communication between our software and the board using parallel and serial ports.

2.3 Tools and Infrastructure

We will be using CVS for our commonly used files.

3The CM Process

3.1 Identification

We need to consider the code, the documentation and the time schedule of our project as the identification items.

Source code: The source code files will be concurrent during the implementation process since we use CVS as our basic CM tool for this identification item. We are developing our software on linux platform, so we will not have much problem about the usage of CVS.

Documentation: We will keep a detailed documentation of the on going development process. The documents will be kept in the CVS in order to make it easier to modify in synchronization with the source code as well as in our shared mail account.

Time schedule: We will also update our time schedule weekly according to each job-package's current status. We will keep it in our mail account as well.

3.2 Management and Control

This part will explain the methods of managing and controlling the changes during the project development.

3.2.1Development

While dividing the modules into teams, we have considered the members' interests, skills and abilities. Although four of the project members have almost same amount of knowledge and technical background, each member feels more comfortable on some parts of the project. Officially each member is responsible for his/her own part of the project, but since our team does not have a large number of members, we can ask to other members for help.

Each member will complete his/her job individually, but changes have to be done through the configuration mechanism to avoid conflicts.

3.2.2Engineering

During the project, we have faced with some technical and engineering problems, and it seems that will face with more of them. As the problem solving mechanism, we have decided to work together. If there is a problem to be solved, first the member who is responsible for that part will analyze the problem and gather the required technical information. After this process, if the responsible member cannot solve the problem, he/she will ask the other members for help.

3.2.3Build

Last term we had divided the project into packages and modules. Also both at the end of the last term and at the beginning of this term, we have prepared a weekly schedule and determined what to do up to the end of this term. According to this carefully prepared schedule, we are now working on our assigned jobs. Every member is doing his/her best to finish his/her job before the given deadline. Also we are combining our finished and ongoing jobs and controlling if we can follow our schedule. By this way, we hope that we will be able to finish our project on time without any problem.

3.2.4Deployment

After finishing project completely, we are going to prepare ready-to-setup software. We are using QT3 as development environment and since QT3 has package deployment tools, we do not expect having problems while preparing this package.

3.2.5Change Requests

During the implementation period, each member may find out some defects on either his/her part or other parts of the project. If this occurs, member will inform the other members what the defect is and how it can be solved. After that if all the members agree with him/her, the required change will be done. The members who are responsible for the part to be changed take over the approved changes, check the baseline CI and apply the approved change to the CI.

3.2.6Defect Tracking

The end of the project period is testing. During the test period, both our team members and some other users who have experience with similar software will use our product and try to find some defects. Every time a defect is detected, the members who are responsible from the imperfect part will solve the problem and guarantee that this change does not affect the other parts of the product.

3.3 Configuration Status Accounting

To synchronize all the members and to manage configuration effectively, up-to-date information should be kept.

Configuration status information is available in CVS and our mail account (groupsosoft@yahoo.com). These documents will consist of source codes, weekly reports, living schedule, design report and change requests. These documents except source codes will be available via our website.

3.4 Auditing

By the help of SCA, we will guarantee that all the functional requirements are met. In order to achieve this, after developing the core parts of the product, we will make a number of demos to our supervisor and according to feedbacks we will review our job and complete the incomplete and incorrect parts. In addition to this, according to the advices of our assistant, we may add some new features to our software.

Also if one of the members requests a change, we are going to review the project and argue if that change is essential.

By doing these, we are going to develop a software which meets all the requirements.

4CM Milestones

We have submitted living schedule and design review to our assistant so we do not repeat it here.

Week Starts	**Our work package to be ready**	**Work to be submitted**
Mar 5	All interface module packages except the debugger-interface	CM Plan, Web page ready
Mar 12	Simulator pic-hex submodule ready	First dev. snapshot demo
Mar 26	All compiler and debugger module packages ready	Possible demo to our assistant

Apr 9	Simulator module's Led, 7-segment, LCD and switch packages ready	Possible demo to our assistant
Apr 30	Upload-to-board module ready	First release and test spec.
May 7	Testing and debugging proccess	Team presentation (exact date not announced yet)
May 14		
May 21		Final package + Final demo/competition
June 11		

5Project Resources

We use CVS as CM tool for our source codes of each module and each member of each team is responsible for CM. We will use the CVS account our department provided us. This tool is enough for keeping the old and new versions of all the software and even the documentation related with this software. We will also use our project website as a tool to keep the latest working version of our software. Our living schedule will be updated weekly (if it needs to be updated).

6Plan Optimization

Being a four member team, each team member should care about maintaining CMP. We will be applying the updates during one or two weeks and the working new version of the overall project should be committed by the end of these periods since we have our milestones then. We may update the plan due to our milestones in case of any inconsistency. The changes in the source code files will be committed to the CVS, so we do not expect any configuration bugs within the work packages. We may only face some problems between the source codes of different modules of which all the group members are responsible to take care.