Name: Rolf Verschuuren
StudentNR: 0916476
Course: DBL 2IO70
Date: 05-02-2015

Abstract of the Project Guide

The subject of the Design Based Learning project “DBL 2IO70” is the design and construction of a machine controlled by a so-called micro-controller. This machine, and micro-controller, are, respectively, a sorting machine capable of sorting black and white discs, and the PP2 Practicum Processor (henceforth referred to as “PP2”).

The goal of this sorting machine is sort a stack of black and white discs into two separate containers, the only requirement is that the machine must contain at least one conveyor belt. The machine is operated by means of two push buttons called “START/STOP” and “ABORT”, where it will start if “START/STOP” is pressed while the machine is in the resting state, if, however, the “START/STOP” button is pressed while the machine is running, it will stop after finishing it's current cycle. If at any time the “ABORT” button is pressed, the machine should stop immediately.

The project consists of multiple phases that have to be completed more or less in order. The first phase is “Orientation” in which every group member has to write an abstract of the “Project Guide”, while the group has to complete a preparatory exercise, and construct a “Work Plan” document. The preparatory exercise consists of a well-defined and relatively small problem, parts of this exercise may be useful later on. The “Work Plan” document details who will be doing what task, and when, assigning responsibility for each task, and setting deadlines for said tasks. The “Work Plan” also indicates which three group members will perform the midterm presentation, and which three (note: the ones who were not involved in the midterm) will preform the final presentation at the end of the course.

The second phase is “Design of the sorting machine”, during this phase a model of the sorting machine is designed and then constructed using the Fisher Technik materials provided by the Tu/e. The design decisions must be properly described and motivated in a document “Machine Design”. To make it clear that the machine will operate as desired, a second outcome of this phase is a description of use-cases, user constraints, and safety properties. A use-case is a description of a usage scenario of the machine, it illustrates the features and observable qualities of the machine. User constraints describe how the user is expected to handle the machine. And safety properties specify a set of relations between inputs and outputs, this is first written in English, but must later be rewritten in UPPAAL. The list of use-cases, user constraints and safety properties together make up the “System Level Requirements” or “SLRs”.

The third phase is “Software Specification”, in this phase a description is written, as accurately as possible, of the required behaviour of the PP2. Described are firstly, which signals from the machine interface are inputs to the program, and what these signals represent. Secondly, which signals from the machine interface are outputs from the program, and what these signals control. And finally, how do the outputs depend on the inputs, that is, what is the required reaction of the PP2 to the inputs. At the end of this phase the document “Software Specification” and accessory

UPPAAL model are done.

The fourth phase is “Software Design”, in this phase a computer program must be constructed in Java. This program must realize the functions specified in the Software Specifications. The correctness of this program must be explained informally. A document “Software Design” must be written, in which the design decisions and choices of the data representation must be explained and motivated.

In the fifth phase “Software Implementation and Integration” the Java program constructed during the “Software Design” phase must be implemented in the Assembly Language of the PP2. For this, a representation in the PP2's memory must be chosen for all variables and data structures from the Java program, as well as a uniform coding standard that describes how common constructs from the Java program are translated into Assembly Language. This data representation and coding standard are collected into a short document “Software Implementation”, handled by the same person who was responsible for the solution of the practicum exercise in phase one.

In the sixth phase “System Validation and Testing”, the aim is to demonstrate that the final product meets the initial requirements. We must also show that the implementation does not either do more than expected, or contains silent code. This can be done in three ways, through code review, where a group of people look at the code together, through test cases, in which a system under test is subject to certain inputs while the outputs are compared to results expected from the specification, and formal proofs, in which properties are proven to hold in all possible states in which the model can be. At the end of this phase a document “System Validation and Testing” must be presented containing all work done in this phase.

After all this is done, the documents resulting from the previous phases must be integrated into a single document “Final Report”. This report also must contain a “Table of Contents”, an “Introduction”, and a “Conclusion”. The “Conclusion” must pay attention to what has been learned during this project, which complications have been encountered and how the group has handled these.

There are some parts of this course that fall outside the phases described above, and are not a part of the document “Final Report”. These are, first, a logbook, that each member has to make individually each week. Second, the allocation of roles such as “President”, “Secretary”, “Quality Assurance Manager”, and “Materials Manager”. Third, a reflection written by each member describing his/her experience in this project. Finally, a mutual evaluation of group members, once halfway through, and another time at the end of the course.