SYSTEM DOCUMENT

Project:

Task:

**Document Version Number**: *(set up a numbering scheme so that you can track versions of the document)*

**Date:**

**Author:**

**Edit History:** (*record who changed the document, when and why)*

*This document is intended to identify and define the solution environment, i.e. the tools you can use and the basic components you have to construct a solution from. You need to identify the capabilities of all parts that you have – both software and hardware.*

*The goal of this document is to know everything you need to about the system to be used for implementing the solution to the design problem.*

*Note that this is going to start as a fairly sketchy document because you don’t know enough. However, it should be iteratively updated as information is learnt about the basic environment in which you have to operate. It should be completed before you can move on to the next stages of the design process..Note – there may be a temptation to repeat information from the Requirements here..* ***you should not do this*** *– instead, you can refer to the appropriate sections of the Requirements. (The reason for this is to try to keep everything synchronized – each item should only appear in one place so that, when an edit occurs, it only has to be edited once otherwise contradictions might appear.)*

1. TABLE OF CONTENTS

2.0 SYSTEM MODEL

(*Once you have fully understood the problem through the Requirements, you can have a first pass at trying to figure out what the basic components are that will be needed. These can be sketched out in a block diagram which is the first step in a Functional diagram of the system –just a sketched out block diagram of the basic functions you think might be needed. It should gradually be updated as the implications of the client requirements become clearer. This model is needed to try to understand how the requirements and solution can be mapped onto the environment you have been given to solve the problem. This may also depend on the human resources- team capabilities – you have available*)

1. HARDWARE AVAILABLE AND CAPABILITIES

(*In the sense of the design problem you have, there are three hardware definitions. First, the Lego components and their mechanical capabilities. The sets limit the structures you can build. What are these limitations? Second, there are the electromechanical limitations. Third there are the electronic/processor constraints (e.g. – how fast can you execute code?). All these constraints need to be identified and listed*.)

1. SOFTWARE AVAILABLE AND CAPABILITIES

(*The software systems you can construct are constrained by the tools and languages you have access to. What Tools are there? What are the advantages and disadvantages of each one – e.g. ease of use versus speed of code execution, generality of operation versus size of code, etc.*.)

4.0 COMPATIBILITY

(*Adherence to requirements within the design environment, e.g. Lego components plug together in certain ways, so everything has to adhere to this. What about software? Are there any compatibility issues there? Will you connect to third party systems, etc.? In this area, you might want to list pieces of code or mechanical structures that have been developed in the lab – these speed up development time but might place constraints on how the system will function, i.e. they were constructed with certain assumptions – you need to know what those were so that you can make sure that they don’t conflict with the current intended usage.)*

5.0 REUSABILITY

(*Structures which may be useful for several parts of the system, existing mechanisms and sub-assemblies which might be used, existing software, etc. The labs might produce some of this – if so, what? – relates to the previous section as well.)*

6.0 STRUCTURES

(*Mechanical structures – reasons for choices, etc., electrical structures, design of software structures, etc. Some of this will derive from the constraints sections earlier. The rationale for choices must be given – this will allow a critical review of decisions, etc., before the design progresses too far or real physical systems are built.)*

7.0 METHODOLOGIES

(*Approaches being taken in all parts of the design and, for software, the basic algorithms to be used. These are really lists of the possible candidate solutions for parts of the problem. They come out of the Ideas Generation phase and will allow a critical analysis before the final design is performed*.)

8.0 TOOLS

(*Details of the tools available to be used to construct the system and why might be useful. Again advantages and disadvantages of the various tools might be given. This might relate back to section 3. If so, do not repeat it here – use a pointer to the information given previously…)*

9.0 GLOSSARY OF TERMS

(*Again, define all the terms you are using so that all members of the Team and any outside readers understand what you are talking about*…)