



STUDENT PLAGIARISM: COURSE WORK – POLICY AND PROCEDURE MTRX 2700 COMPLIANCE STATEMENT INDIVIDUAL / COLLABORATIVE WORK

I/We certify that:

- 1. I/We have read and understood the *University of Sydney Student Plagiarism:* Coursework Policy and Procedure;
- I/We understand that failure to comply with the University of Sydney Student Plagiarism: Coursework Policy and Procedure can lead to the University commencing proceedings against me/us for potential student misconduct under Chapter 8 of the University of Sydney By-Law 1999 (as amended);
- The Work undertaken in this course is substantially my/our own, and to the extent that any part of this Work is not my/our own I/we have indicated that it is not my/our own by Acknowledging the Source of that part or those parts of the Work.

Name	Signature	SID	Date
Melissa Mitrevski		440207636	05/06/2015
Meg Flannery		440291196	05/06/2015
Lydia Drabsch		311217591	05/06/2015

GROUP CERTIFICATION

We certify that the design, implementation, and documentation presented by our Group are, unless otherwise acknowledged, the work of our group members only, and that the individuals named below were responsible for completing the following percentages of the modules named.

Name	%	of the Module	Signature
Lydia DRABSCH			
311217591			
Melissa MITREVSKI			
440207636			
Meg FLANNERY			
440291196			

CONTENTS

1	Intr	troduction			
	1.1	Document Identification			
	1.2	System Overview			
	1.3	Document Overview			
	1.4	Reference Documents			
		1.4.1 Acronyms and Abbreviations			
2	Syst	tem Description 1			
_	2.1	Introduction			
	$\frac{2.1}{2.2}$	Operational Scenarios			
	$\frac{2.2}{2.3}$	System Requirements			
	$\frac{2.5}{2.4}$	Module Design			
	2.5	Module Requirements: Module X			
	2.0	2.5.1 Functional Requirements			
		2.5.1.1 Inputs			
		2.5.1.1 Inputs			
		2.5.1.2 Process			
		2.5.1.4 Timing			
		G C C C C C C C C C C C C C C C C C C C			
		2.5.2 Non-Function (Quality of Service) Requirements			
		2.5.2.2 Interfaces			
	0.6	2.5.2.3 Design Constraints			
	2.6	Conceptual Design: Software Module X			
		2.6.1 Assumptions Made			
		2.6.2 Constraints on Module X Performance			
3	\mathbf{Use}	r Interface Design 3			
	3.1	Classes of User			
	3.2	Interface Design: User Class Y			
		3.2.1 User Inputs and Outputs			
		3.2.2 Input Validation and Error Trapping			
1	Har	rdware Design 4			
-	4.1	Scope of the Something System Hardware			
	4.2	Hardware Design			
	1.2	4.2.1 Power Supply			
		4.2.2 Computer Design			
		4.2.3 Sensor Hardware			
		4.2.4 Actuator Hardware			
		4.2.5 Operator Input Hardware			
		4.2.6 Operator Output Hardware			
		4.2.7 Hardware Quality Assurance			
	4.3	Hardware Validation			
	$\frac{4.3}{4.4}$	Hardware Calibration Procedures			
	$\frac{4.4}{4.5}$				
	4.0	Hardware Maintenance and Adjustment			

5 Software Design				5
	5.1	Softwa	are Design Process	5
		5.1.1	Software Development Environment	5
		5.1.2	Software Implementation Stages and Test Plans	5
	5.2	Softwa	are Quality Assurance	5
	5.3	Softwa	are Design Description	5
		5.3.1	Architecture	5
		5.3.2	Software Interface	5
		5.3.3	Software Components	5
	5.4	Preco	nditions for Software	6
		5.4.1	Preconditions for System Startup	6
		5.4.2	Preconditions for System Shutdown	6
6	Svs	tem P	erformance	6
	·	6.0.3	Performance Testing	6
		6.0.4	State of the System as Delivered	6
		6.0.5	Future Improvements	6
7	Safe	etv Im	plications	6
8	Cor	ıclusio	ns	6
9	Apj	pendix		7
_				
L	IS	ГОЕ	FIGURES	
L	IS	ГОЕ	F TABLES	
	1.1	Acron	yms and Abbreviations	1

1. Introduction

1.1 Document Identification

This document describes the design of something. This document is prepared by Group ZULU for assessment in MTRX2700 in 2010.

1.2 System Overview

A brief statement of the purpose of the system or subsystem to which this document applies.

1.3 Document Overview

A short road map of the document, to provide an orientation for the reader. Summarise the purpose and contents of this document.

1.4 Reference Documents

The present document is prepared on the basis of the following reference documents, and should be read in conjunction with them.

- . CAN Specification 2.0. Robert Bosch GmbH, Stuttgart, September 1991.
- . OMG Unified Modeling Language Specification, vers.1.3, June 1999.

1.4.1 Acronyms and Abbreviations

Table 1.1 lists the acronyms and abbreviations used in this document.

Table 1.1: Acronyms and Abbreviations

Acronym	Meaning	
MPC	Model Predictive Control	

2. System Description

This section is intended to give a general overview of the basis for the something system design, of its division into hardware and software modules, and of its development and implementation.

2.1 Introduction

Give a technical description of the function of the whole system, in terms of its constituent parts, here termed modules. Generally, a module will have hardware and software parts.

2.2 Operational Scenarios

Describe how the system is to be used. There may be several different ways that it can be used, perhaps involving different users, or classes of user. Present use case diagrams here if you are using them. Each operational scenario is a part through a use case diagram—a way of using the system,

with different outcomes or methods of use.

You should also consider the various failures that may occur, and the consequences of these failures.

2.3 System Requirements

The operational scenarios considered place certain requirements on the whole something system, and on the modules that comprise it.

Statement of requirements that affect the system as a whole, and are not restricted to only a subset of its modules.

2.4 Module Design

Describe the breakdown of the design into functional modules. Each module probably contains both software and hardware.

Then include a section like the following (2.5) for each module. Not all of the sub-headings may be relevant for each module.

2.5 Module Requirements: Module X

The operational scenarios considered place certain requirements on the something system, and on the modules that comprise it.

2.5.1 Functional Requirements

This section describes the functional requirements of Module X those requirements that must be met if the module (and system) is to function correctly.

2.5.1.1 Inputs

Describe each external input, including signal encoding and timing, message encoding and timing, protocols, file formats, protection against input errors, etc, as relevant.

2.5.1.2 Process

Describe the internal signal transformations and/or computer processing functionality required within the module, required performance limits, and error tolerances as appropriate.

2.5.1.3 Outputs

Describe outputs that must be produced for the module to function correctly, including timing, frequency, protocols, etc as relevant.

2.5.1.4 Timing

Any required timing or latency specifications that must be met.

2.5.1.5 Failure Modes

Required functionality (if any) in the event of failure of various nominated components.

2.5.2 Non-Function (Quality of Service) Requirements

Non-functional requirements do not need to be met for the device to have basic function, but are required to provide specific levels of performance or engineering quality.

2.5.2.1 Performance

Requirements such as computational loop time, accuracy, etc.

2.5.2.2 Interfaces

Requirements such as computational loop time, accuracy, etc.

2.5.2.3 Design Constraints

Practical or commercial considerations, such as programming languages, processor or other hardware, etc.

2.6 Conceptual Design: Software Module X

Now, for each module, give the outline of how it will work. In this section it is appropriate to present

The rationale for the design decisions that were made why things were designed the way they were block diagrams,

mathematical models and algorithms,

data flow diagrams

state-transition diagrams

listings of input and output formats

listings of message and data formats

responses to identifiable error conditions

responses to identifiable failure conditions

as appropriate for each module.

2.6.1 Assumptions Made

State any assumptions made.

2.6.2 Constraints on Module X Performance

State any constraints that may prevent the design from satisfying its requirements.

3. USER INTERFACE DESIGN

Give a detailed description of the design of the user interface. This will give the reader a good view of how the system functions from the users perspective.

3.1 Classes of User

If there are different user interfaces presented to different classes of users, define these user classes, and how access by the various user classes is enabled or disabled.

3.2 Interface Design: User Class Y

3.2.1 User Inputs and Outputs

Description of how the user presents inputs to the system, and how the system responds to those inputs. Include a description of how the user knows the state of the system.

3.2.2 Input Validation and Error Trapping

Describe how the system validates user input, and how operator errors are trapped and can be recovered from.

4. HARDWARE DESIGN

Give a detailed description of the design of hardware. The description should include mechanical drawings, location diagrams, electrical circuit schematics, circuit simulation or test results, PCB overlays, wiring diagrams, connector pinout lists, pneumatic/hydraulic circuit diagrams

4.1 Scope of the Something System Hardware

Statement of what is, and what is not, being designed and described here.

4.2 Hardware Design

4.2.1 Power Supply

Power supply method and rating, fusing, distribution, grounding and protective earth as appropriate.

4.2.2 Computer Design

Description of computer hardware, including all interface circuitry to sensors, actuators, and I/O hardware.

- 4.2.3 Sensor Hardware
- 4.2.4 Actuator Hardware
- 4.2.5 Operator Input Hardware
- 4.2.6 Operator Output Hardware
- 4.2.7 Hardware Quality Assurance

Describe any measures that were taken to control (improve) hardware quality and reliability Heart-beats, brownout conditioning/resets, reset conditions, testing and validation, etc.

4.3 Hardware Validation

Details of any systematic testing to ensure that the hardware actually functions as intended.

4.4 Hardware Calibration Procedures

Procedures for calibration required in the factory, or in the field.

4.5 Hardware Maintenance and Adjustment

Routine adjustment and maintenance procedures.

5. SOFTWARE DESIGN

The software requirements and overview have been dealt with elsewhere in this document. The present section addresses the design and implementation of the software that forms the X system.

5.1 Software Design Process

How you went about designing the software top down, bottom up, OOD, functional view, etc.

5.1.1 Software Development Environment

The tools that were used, both software (compilers, assemblers, etc) and hardware (development boards, etc.).

5.1.2 Software Implementation Stages and Test Plans

Describe the way you went about implementing the software - staged implementation, pseudocode (PDL), unit testing procedures, integration testing, etc. Identify dependencies e.g this had to be done before that, etc.

5.2 Software Quality Assurance

Describe any measures that were taken to control (improve) the software quality code or documentation standards, code walkthroughs, testing and validation, etc.

5.3 Software Design Description

5.3.1 Architecture

Describe the high-level architecture of the software that is, the top-level flow of control, and how the various functional modules communicate.

In this section, you can put state transition diagrams, sequence diagrams, etc.

5.3.2 Software Interface

Describe the public interface of each software module.

5.3.3 Software Components

This is a detailed view of the internal workings of each of the software modules.

5.4 Preconditions for Software

5.4.1 Preconditions for System Startup

Describe any preconditions that must be satisfied before the system can be started.

5.4.2 Preconditions for System Shutdown

Describe any preconditions that must be satisfied before the system can be stopped.

6. System Performance

6.0.3 Performance Testing

Give the results of testing conducted to determine the characteristics and performance of the system - memory usage, loop time, system accuracy, repeatability, ease of use, etc.

6.0.4 State of the System as Delivered

A statement of your groups opinion of the conformance of the system with the specification.

6.0.5 Future Improvements

Present a prioritised list of improvements to be made in future releases, giving reasons for the improvement and priority rank.

7. SAFETY IMPLICATIONS

By law (NSW Occupational Health and Safety Act 2000; NSW Occupational Health and Safety Regulation 2001) all employees who design plant, machinery or equipment must identify foreseeable safety hazards associated with the equipment, and then assess and control the identified risks. Although this law does not apply directly to student designs, you should consider it here.

8. Conclusions

9. APPENDIX