



MODULAR PROGRAMME
ASSESSED COURSE-WORK SPECIFICATION

Module Details:

Module Code: UFMEMY-20-3	Module Title: Embedded Co-design with C & VHDL	
Module Leader: Nigel Gunton		
Module Tutors: Nigel Gunton Craig Duffy		
Assignment CW1	Element Number: Weighting 1 : 60%	Total Assignment Time: 50 hrs + lab time

Dates:

Date assignment issued to students: November 2010	Date for return of marked work: 20th May 2010
Submission Place: Project Room - 2Q30 (Open 09.00 - 18.00)	Date of Submission: 31st March 2011
	Time of Submission: 14.00

Deliverables:

As listed on the Assignment spec sheet

Overview :

This assignment will provide experience of the problems and decisions in developing co-design projects. There are many possible solutions to the design problems depending on the way in which you choose to partition each problem.

The marks shown below are indicative and may be subject to modification during the year. The recommended target hardware is the TS7300 embedded board. This provides the opportunity to implement low-level, interrupt driven, device drivers along with the custom hardware and a more sophisticated user interface. Incomplete solutions may be acceptable on this platform due to the greater complexity, but discuss with your lab tutor. You may choose to use the older Excalibur boards for the assignment, however in this instance you will be expected to deliver a fully working system.

You will develop a portfolio of documentation and results which will comprise the assignment deliverables. The deliverables are further defined below. Marks will be allocated across all design problems. To obtain 40% you would have to have completed an analysis of the design problem along with implementation of at least part of the software/hardware required. To obtain 70%++ you would have to have completed most of the requirements and made significant progress in the others, the more progress, the greater the ++. The exact marks attribution is subject to variation due to the inherent difficulties in working with temperamental hardware.

1. Design Problem :

To design and implement a 'pick and place' system. A robot arm will be trained to repeatedly move an item from one location to another. The training and control input will be from a 4x4 key-pad. The arm has four independent servomotors. The software and hardware control will be implemented on either the TS7300 or the Altera DE0. A different approach will be required for each board. The work will be done in teams of 4. You will be required to partition the problem into suitable hardware and software units, integrate these units into a complete system and test. The documentation should show justification for any design decisions that you make as well as development logs for both hardware and software.

At a minimum your documentation should include the following:

- An overview of your complete system including justification for design decisions including hardware/software partitioning .
- A list of the team members along with their roles and responsibilities, (to include authorship details for hardware & software components).
- An appendix containing your Standards Reference Manual.
- Verification and testing strategies with results.
- Schematics and code outlines (VHDL, Verilog or C) for your components.

Possible extras could include

- Data sheets for all your custom hardware components.
- 'README's and/or man pages for any device drivers, custom libraries, system software or HALs¹ that you have developed.
- User Manual for your system or a 'White Paper' extolling the virtues of your design.

¹ Hardware Abstraction Layer

TS7300 Board:

This board has an ARM 9 based microcontroller and a Cyclone II FPGA. There is an interface between the 2 devices. The ARM is running Debian GNU/Linux. The design requirements are the same, however you have full O/S support for the software. You are still required to use the keypad for input. You are strongly encouraged to use the TS7300.

Altera DE0 Board:

This board has a Cyclone III FPGA fitted. This supports a 'soft-core' processor integrated with custom hardware. Using the Nios2 softcore CPU as a base you will implement a system to control a robot arm. This is a brand new solution and consideration will be given for attempting this. There is the potential to use a small embedded O/S, FreeRTOS, or to write low level C for this solution.

The Robot Arm :

An image and overview of the servo operation can be found at

http://www.cems.uwe.ac.uk/~ngunton/images/open_day_servo.pdf

The Key-Pad :

Data-sheet details for the keypad at

<http://www.cems.uwe.ac.uk/~ngunton/images/keypad.png>

The Design Problem :

The goal of this section is to develop a 'pick and place' system with training capabilities. A robot arm will be driven by manual control in order to define the waypoints for the place and pick. Once trained, the arm should automatically follow the programmed path on receipt of a run command.

The system must be a stand-alone system that does not rely on a network connection or other remote access such as serial comms. All input to the system must be via the 4x4 keypad, except during development. The outputs of the system are restricted to a small LCD screen and the servo connections. You may use any available onboard memory. You may also use the two leds on the TS7300 board for status/configuration/error signalling. Similarly with the DE0 board, you may use the additional seven-segment displays, leds etc.

Using IP.

The use of pre-existing software or softcore IP is permitted. Any IP so used must be clearly attributed. In this instance the deliverables must include documentation explaining the function of the IP as well as any modifications that may have been made as part of the integration process.

Team Meetings

You are strongly advised to follow the team meeting guidelines and procedures outlined in the two documents available from the module web page.

Deliverables :**NOTE:**

Marks distribution is indicative of the relative weighting of the components within this section but may be subject to change in the event of unexpected technical problems.

All the marking will be done in line with the guidance given in http://www.cems.uwe.ac.uk/~ngunton/ufmemy_marking_grid.doc

- Design and development documentation which must include
 - Discussion of design partitioning and justification for your partitioning in problem 1. 15%
 - Your final hardware design, the marks indicated are shared between the problems.
 - i) Design 10%
 - ii) Implementation of components 10%
 - Your final software design, again marks are shared.
 - i) Design 15%
 - Change logs for Hardware and Software development 5%
 - Verification and Testing
 - i) Strategies for individual components and whole system. 10%
 - ii) Results 5%
- A viva and lab demonstration of the components and system for the problem that you have achieved the most with. 30%

NOTE:

The lab demos etc will take place during the last 2 lab sessions of the semester or sooner by arrangement.