MANAGING AN EFFECTIVE HARDWARE BASED LARGE CLASS ASSIGNMENT

David Edwards[1]

Abstract *¾ Students want assignments that are interesting, relate to the real world, not all be due near the end of the semester, and are graded and returned quickly. From the instructor's viewpoint assignments should encourage student learning, be achievable by all students, extending the better student, and be easy to grade. This paper describes the assignment paradigm used in large class computer engineering course. Three progressive assignments build to a real hardware task - controlling a Bilby 'maze mouse'. The first two assignments are subsets of the third performed on a computer simulator. Concepts developed for each assignment can be used in the next. The final assignment features the inclusion of optional 'bonus' sections for more marks. Assessment by demonstration of the working assignment and the use of a Marking Criteria Sheet for the written report leads to a dramatic reduction in marking time while improving student feedback.*

Index Terms ¾ Assessment, Computer engineering, Hardware assignment, Problem-based learning.

INTRODUCTION

At the Gold Coast campus of Griffith University, computer engineering is introduced in the first semester of the first year of both the Bachelor of Electronic Engineering (BEng) and the Bachelor of Information Technology (BIT) programs through a course entitled 'Microprocessors'. This course had an enrolment of 185 in 2002. Within the course students are introduced to the concepts of a microcomputer and a microprocessor development system.

Students come to the course with a wide range of programming abilities and interests. Most have no prior experience of assembly language programming. The challenge for the course convenor was to implement an active learning [1] assessment structure that generated interest, was challenging for better prepared students, yet was achievable by all.

Rather than have a series of programming exercises, the decision was made to structure the course around a major hands-on real hardware assignment. The hands-on task chosen was to program a small two wheeled robot, called a Bilby, to follow a path around a circuit. The major task to be achieved by the end of the course gave an immediate relevance to the course material. The assignment task generated interest in the students, gave them a sense of achievement when completed, promoted connected learning amongst the students [2], and enhanced achieving the course objectives. As well, in these days of diminished resources, the assignment was efficient in resource utilization.

Assignment work thus became the major focus of the course. The problem to be overcome was how to prepare students progressively for this major task. Most courses develop their material progressively and students are only prepared for a major activity in the last few weeks at the end of the course. The solution chosen was to split the major task into three progressive tasks. In this way students could undertake the first assignment relatively early in the course. Although a 'simple' task it had real relevance as it is part of the larger task.

However a potential problem with a phased or progressive assignment is encouraging and supporting the students cannot get out the first or second stages. Are they then unable to progress to the next stage? Less well prepared students may take longer to develop a suitable level of programming skill. The solution adopted was to provide, as feedback, model solutions for the earlier assignments which students could use.

Having a progressive assignment meant that feedback on performance to students had to be quite rapid, essentially within a week. This lead to the development of assessment and marking strategies that were time efficient while still providing useful feedback to students.

THE COURSE

The Microprocessor course is a one-semester course that carries 25% of the weight for the semester. It forms a foundation for further computer engineering studies for the BEng students. It is a one-semester course that carries 25% of the weight for the semester. For the BIT students the use of simple development system monitor routines introduces the functional requirements of operating systems; the use of a simple assembler introduces concepts of compilers used in later courses. The constraints of assembly language programming introduce students to the need to carefully plan and document programming tasks.

The flow of the content has been structured around the assignments. To allow students to progress with assembly language programming skills sufficiently quickly to be ready to undertake their assignments, the course content is taught in lectures with a 'just in time' approach. The student

[1] David Edwards, Griffith University, School of Engineering, PMB 50, Gold Coast Mail Centre, 9726 Queensland, Australia d.edwards@mailbox.gu.edu.au

contact time for the course is 3 one-hour lectures per week and a 1hour small group tutorial. The lectures are supported by a suite of course resource materials which are organized both in a 'just in time' manner and in a more formally structured 'just in case'.

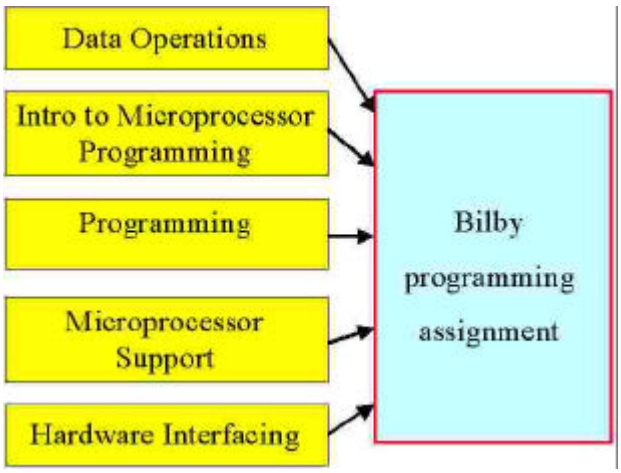

FIGURE. 1
COURSE CONTENT MAP.

In addition to the assignments, the course is assessed by mastery tests after each of the 5 course content modules.

COURSE RESOURCES

The course is supported by a printed set of course notes and a course website, running under the Blackboard learning resource management environment [3], that is organised around the Bilby assignment.

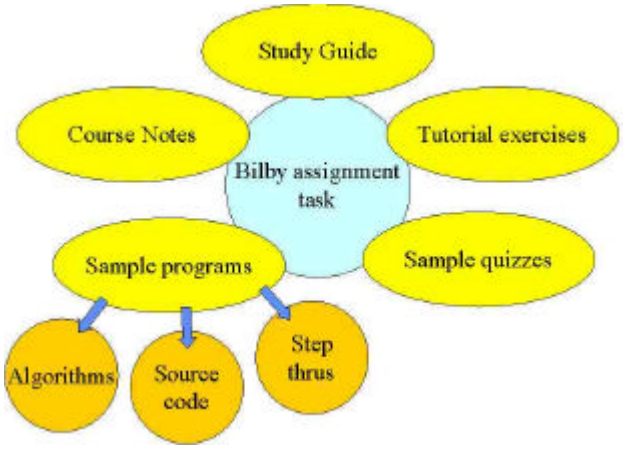

FIGURE. 2
OVERVIEW OF WEBSITE MATERIALS.

Not shown in the structure in Figure 2 is the supporting set of resources on the website. These include a noticeboard, shown on log-in, for announcements; a set of lecture notes (in PowerPoint format) posted after each lecture; sample solutions to assignments 1 and 2 posted immediately after expiry of due date; answers to mastery tests; and the latest students' marks. Under Griffith University policy, marks for each assessment item should be published. In the published list, students are identified by their student number (which is never published along with their name) and, for this course, the list is sorted by total mark so far. This sorted display has proved popular with the higher achieving students as they try and out perform each other.

Other resources support the actual assignment task. Although the major assignment task involves hardware programming, the earlier assignments are completed using an in-house developed program, EaSim, which simulates the operation of the hardware. EaSim is available from campus computing laboratories and students are permitted to make a copy of the program for their personal use. EaSim is also used as the development platform for the final assignment.

THE ASSIGNMENT TASK

The assignment task was chosen:

- to require a range of programming skills,
- to have a range of I/O tasks,
- to involve using a real piece of hardware,
- to be capable of being broken up into a number of stages
- to have a main task that all students should be able to achieve while having a number of optional tasks of graded difficulty, and
- to be interesting as possible to the students.

In 2002, the main assignment task was to use the 68HC11 based HandyBoard to control the operation of a small two-wheeled robot. The small robot is called a 'Bilby' after an endangered Australian small native marsupial that is attracting some popularity as an alternative to the Easter Rabbit.

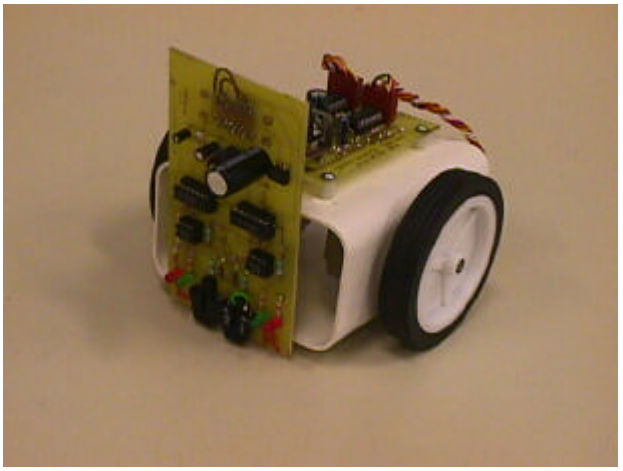

FIGURE. 3
THE 'BILBY' TWO WHEELED ROBOT – FRONT VIEW.

For propulsion, the Bilby has two stepper motors each directly driving a wheel. The stepper motors are controlled by direction and clock signals sent to each wheel. The Bilby can travel forward or backwards, and turn either to the left or right.

Two Infrared transmitter–receiver pairs mounted at the front of the Bilby can each detect whether the track under each sensor is black or white.

The Bilby is required to follow a black track painted on a white background. The track is slightly wider than the spacing of the sensors so when on the track both sensors are over the track. If the Bilby is placed off the track, it should reverse up to find the track. There are a number of optional tasks that can be undertaken including implementing a variable speed control, sounding a warning buzzer before reversing, implementing 'turn indicators', and handling the problem of smoothly rounding a gentle bend.

The program that that controls the Bilby operation has to output appropriate direction status messages to HandyBoard screen, have simple 'password' protection, have the Bilby stop and start on pushbutton command, and generate the clock and direction control signals for the stepper motors.

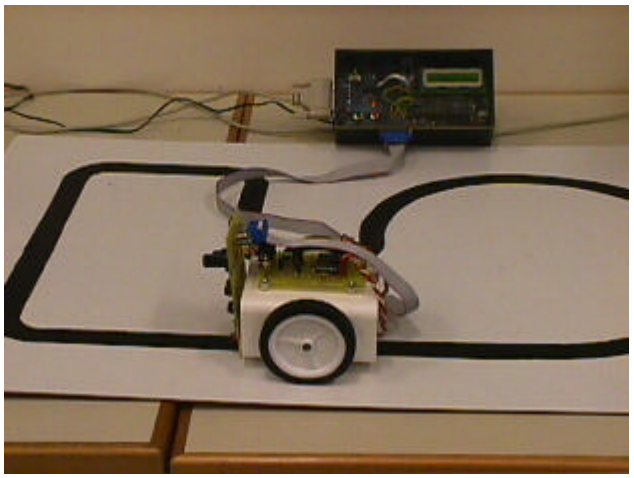

FIGURE. 4
THE 'BILBY' FOLLOWING A TRACK.

Two earlier assignments are wholly simulator based involving sub-tasks of this assignment. As students are given all three assignments tasks at the beginning of the subject, they could see the relevance of the simplified early tasks. The major assignment is organised as a base task that all students should be able to achieve. Completing this task to a satisfactory standard carries 60% of the allocated marks. In addition there are six optional, or bonus, tasks each worth 10%. A student who attempts all options can achieve more than 100% on the assignment.

ASSESSING THE ASSIGNMENT

As in-semester assessment is intended to be formative as well as summative, it is imperative to provide the feedback as rapidly as possible. This is particularly the case with the assignments where each assignment leads from the previous another. Given the need to fit 3 assignments in a 13 teaching week, and that the students do not reach a knowledge stage for a useful assignment until about week 4, there is only 4 weeks between submissions of assignments 1, 2 and 3. In practice, for useful feedback, assignments have to be returned within 1 week of submission.

Programming assignments are normally quite time consuming to assess. Whether the program does the task has to be assessed as well as the quality of programming and the associated documentation. The assignments are treated in the manner of a professional contract: meet the specifications and receive full payment, breach a clause and penalties apply.

The decision was made to separate the assessment of the program (Does it do the task?) and the documentation. All assignments are assessed in two stages. The running program is demonstrated and marked for achievement of specifications in lab/tutorial time, and then the write up is assessed separately. The use of a printed marking sheet/marking scheme means the 185 students can be assessed for an assignment in 10 hours of lab/tutorial time and about 3-4 hours of report marking time. Student results are normally posted with an hour of the assignment deadline.

Assessing the program

Students are told that it is very important that their program actually meets the specifications. Emphasis is placed on the program 'working' rather than being 'pretty'. Students when they start assembly language programming are often very nervous and unsure of themselves. By not assessing 'the quality' of their program, they are encouraged to get the task done. In this way almost all will achieve a program that is at least 90% correct according to the specifications. To encourage innovative thinking about algorithms and coding, bonuses are given for the shortest executable code. These bonuses encourages students to explore the 68HC11 instruction set for alternate ways of doing things.

The assessment philosophy is that a program must meet a minimum acceptable standard. If a program meets these specifications then the student will receive 100% for demonstration part of the assignment. If they fail to meet these standards, marks will be deducted – penalties will be applied. The penalties applicable to any breach are detailed on the assignment marking sheet (see Appendix 1).

Students are required to demonstrate their working program in their normal weekly tutorial/laboratory timeslot. The operation of the program is demonstrated to the class tutor on an individual basis. With the marking sheet listing the task specifications, the actual assessment takes the tutor

only a minute or two for each student. The tutor records no marks, but provides oral feedback. If the program does not meet some specification, that is noted on the marking sheet and a penalty is recorded. After the tutor has assessed the demonstration the marking sheet is signed off. Each student is responsible for keeping their marking sheet and submitting it as part of the documentation. It is quite possible to assess the 20 students in a tutorial class well within the 55 minute tutorial time.

To encourage students not to leave their demonstration to the last moment, a 2 mark bonus applies to programs demonstrated a week, or more, before the deadline. This 2 mark bonus, being 20% of the marks for the each of the first two assignments, acts as a strong encouragement for early demonstration.

Late assignments may be demonstrated in special 'catch-up' timeslots before the due time of report submission.

Assessing the documentation

Students are told that their report must meet a minimum acceptable standard. A sample standard report is provided to the students with their course documentation. If their report meets the minimum standard then they will receive 100% for documentation section of the assignment. If they fail to meet this standard, marks will be deducted – penalties will be applied. The penalties applicable to any breach are detailed on the assignment marking sheet (see Appendix 1).

The deadline for the submission of the documentation section of the assignbment is the Friday of the 'demonstration week'. No extension of time is normally given as model answers for the assignment are released on the website that evening. The assignment documentation requirements are very tightly specified even as to the order of the various sections. The assignment marking sheet is to be included as the final page.

The tight specifciation of content makes the marking task much simpler. Each section should be found in the same order. Having the marking sheet at the back makes it very easy to access to record feedback. Feedback is given by brief comments in the text of the assignment as well as by noting unsatisfactory aspects, and the appropriate penalty, on the marking sheet. All assignemnts are marked by the course convenor. This task takes approximately 3-4 hours per assignment. Marking usually starts about 2pm on the deadline day and is completed within half an hour of the 5pm deadline. The rapid marking is made possible by the marking scheme and the fact that most of the students will met the satisfactory standard for their documentation.

STUDENT FEEDBACK

Students are surveyed at the end of each course offering for feedback. The survey for this course in done at the start of a test in the final week. This ensures a 95% response rate.

Students are asked whether they thought the assignments contributed to their learning, and for the best and worst things about the assignments. A summary of typical 2002 student responses in given in Table 1. Informal feedback, reported by all class tutors, is the obvious sense of achievement shown by students when their Bilby runs. This is reflected in the student comments.

TABLE I
SUMMARY OF STUDENT RESPONSES

Best feature was making a Bilby run.
Practical course – Hands on programming
It was great breaking down one large task into 3 assignments because you could see how each assignment related to the overall task.
Making a robot move was excellent.
The opportunity to use software on a real life situation.
Seeing the program run on hardware.
I felt I learnt more and retained it by doing it in stages.
Programming the robot and seeing it move.
The building process where each assignment leads to the next.
The satisfaction of doing it properly.
The three separate progressive assignments.

The few negative responses concentrated on:

- The workload involved in the assignments. A few thought that the assignments took too long to do.
- The lack of access to the hardware. The hardware was only available for the two weeks before assignment 3 was due. The lab was opened many extra hours for testing of assignments in the two weeks before the due date. Unfortunately many students, in spite of repeated warnings to the contrary, tried to leave their testing until the final week.
- The scheduling of some of the tutorial classes meant that some students had more time for testing than others.

CONCLUSION

By structuring the course around a major 'hands-on' assignment a relatively large enrolment first year course has engaged the students in active learning. By adopting suitable streamlined assignment marking and feedback techniques, turn around time has been minimised and staff load has been kept to a minimum. This has been done without disadvantaging student learning.

ACKNOWLEDGEMENT

Thanks are due to my colleague, Charles Hacker, who developed the program used to download S19 object code files to the HandyBoard for testing. Charles also wrote the Buffalo work-alike operating system for the HandyBoard. Thanks also to Gilles Ravanelli, the electronics technical officer supporting this course, who built up the HandyBoards for us as well as designing and manufacturing the Bilby hardware.

REFERENCES

[1] Azemi, A, "Developing an Active Learning Environment with Courseware Approach", *Proceedings of the FIE conference*, 1997.

[2] Tinto,V, "Colleges as Communities: Taking Research on Student Persistence Seriously", *The Review of Higher Education*, Vol 21, No 2., 1998, pp. 167-177

[3] http://www.blackboard.com/

APPENDIX 1

Specification for Assignment 3

'Your microcontroller is required to control the motion of a Bilby. A Bilby is a two-wheeled (and a skid) mobile robot. Movement of the Bilby is achieved by driving either, or both, of its stepper motor driven wheels.
The Bilby is required to follow a path set by a black line on a white background. Following the path is controlled by two infrared transmitter-detectors which in normal operation sit above the black line and return '0's. If the Bilby moves too far to the left, the left hand detector moves 'off' the line and its output changes to a '1'. If the Bilby moves too far to the right, the right hand detector moves 'off' the line and its output changes to a '1'. The outputs from the left hand and right hand detectors are returned as bits 4 and 5 respectively of the digital inputs.
The Bilby is also controlled by two use-operated pushbuttons on the HandyBoard. When the Bilby is stopped, pressing the 'Start' button should activate the Bilby, setting it to follow the path. Pressing the 'Stop' button should stop the Bilby.
The Bilby is initially set along the 'track'. When the word 'GO' is typed at the keyboard, the Bilby is able to move (forward, stopped, or reverse) according to the controller button settings.
The Bilby should follow the track, the signals from the IR units being used to decide whether the Bilby should be moving straight ahead, or turning left or right.

When the Bilby is traveling straight ahead, the HandyBoard LCD display should show 'FORWARD' in the 7 leftmost places on the top line. When it is turning right, the display should show 'RIGHT>' in the 6 rightmost places on the bottom line, when it is turning left, the display should show '<LEFT' in the 5 leftmost places on the bottom line.
When the Bilby is stopped, the LCD display should show 'Stopped' in the 7 rightmost places on the top line. '

This basic task carries a maximum mark of 12. To achieve higher marks any, or all, of the following Option sections may be attempted. Successful implementation of each option will carry a bonus of two marks.

Option 1: If the Bilby leaves the track such that neither sensor is over the marked track then the Bilby reverses until it returns to the track. It then proceeds to follow the track. When the Bilby is reversing, the display should show the fixed message 'Reversing' in the centre of the top line of the LCD display.

Option 2: The speed of the Bilby is to be controlled, while the Bilby is moving, by the rotary knob on the front of the HandyBoard. The speed of the Bilby should change from the slowest speed at which the Bilby will move to the fastest speed at which the Bilby will move.

Option 3: *Before* the Bilby reverses it is to issue a warning. The warning consists of a 1/2 second beep, followed by a 1/2 second silence, followed by a 1/2 second beep. The beep consists of the HandyBoard buzzer sounding at approximately 500Hz for the 1/2 second. After the warning the Bilby should move in reverse.

Option 4: While the Bilby is turning a 'gentle' left hand bend on the track it will make a mixture of steps to the left and steps straight ahead. This will cause the turn messages on the LCD screen to switch between FORWARD and <LEFT. Your program is to ensure that the message stays as <LEFT and FORWARD does not appear. A similar situation is to hold when the Bilby is rounding a 'gentle' right hand bend.

Option 5: When the Bilby is rounding a left hand bend (gentle or sharp), the top LH green LED on the HandyBoard should flash. When the Bilby is rounding a right hand bend, the top RH red LED should flash. The flash rate for both LEDs should be once per second. The flash rate may vary as the Bilby speed is varied.

Bonus: If the working assignment is demonstrated in normal tutorial time at least one week before the 'due date' tutorial then a bonus of 2 marks will apply.

APPENDIX 2

Assignment 3 Marking Scheme

Student Name:	Student No.:

Attach this sheet to your assignment as the **last** page.
This assignment will be marked according to the following schedule:

SECTION	**YES**		'Bonus'
Demonstration of working program at least 1 week before 'Demo' deadline		**Bonus** of 2 marks	
Does Bilby reverse when off the track?		up to 2 marks	
Is variable speed running implemented?		up to 2 marks	
Does warning sound before reversing?		up to 2 marks	
Is LCD display 'stable' when rounding bends?		up to 2 marks	
Do turn indicator LEDs flash correctly when rounding bends?		up to 2 marks	
		Total Bonus	
SECTION	**YES**	**NO** Potential Penalty	Penalty
Late demonstration of working program after deadline but before cut-off?		2 marks per day	
Is GO read correctly?		up to 2 marks	
Is LCD display as specified when running?		up to 3 marks	
Does Bilby start and go straight ahead correctly?		up to 3 marks	
Does Bilby follow bends correctly?		up to 3 marks	
Does Bilby turn correctly according to IR detectors?		up to 3 marks	
Tutor's Signature		Subtotal (Penalties)	
Description of algorithm included? Is it clear and easy to understand?		5 marks up to 5 marks	
Is a copy of the assembly language program included? Is code carefully documented? Are labels used in ARG column? Is the output formatted neatly? Is code, or algorithm, elegant?		5 marks up to 5 marks up to 2 marks up to 2 marks up to 3 marks	
Is the assembler output included and formatted neatly?		up to 2 marks	
Is the Object Code included and formatted neatly?		up to 2 marks	
Assignment has cover sheet, marking sheet, and is correctly stapled?		up to 5 marks	
Assignment **not** demonstrated in tutorial class		10 marks	
TOTAL of any penalties			

Assignment Mark (**12** + bonuses - any penalties) **.....**