Int. J. Elect. Enging Educ., Vol. 19, pp. 357–366. Manchester U.P., 1982. Printed in Great Britain

AN ADVANCED ELECTRONICS TEACHING LABORATORY

R. M. F. GOODMAN, G. E. TAYLOR and A. F. T. WINFIELD
Department of Electronic Engineering, University of Hull, England

1 INTRODUCTION

This paper describes a new third-year teaching laboratory, which has been implemented during the current academic year, as an integral part of the four-year honours degree course in Electronic Engineering at the University of Hull.

Formal teaching and laboratory work in the department are continually reviewed, as befits the rapidly changing nature of the subject. As part of this review process, the third year laboratory was closely examined in the summer of 1981, and it was felt necessary to make a number of major changes to reflect both the evolving nature of the technology, and changes in the philosophy of engineering education[1].

This paper proceeds as follows: the next section examines in more detail the reasons for change, and this is followed in section three by a brief examination of the actual laboratory and its experiments. Next, we examine the mode of assessment adopted for the laboratory, and, finally, conclude by reflecting upon the first year of operation.

2 REASONS FOR CHANGE

Traditionally, electronics teaching laboratories have been based upon formal experiments designed to demonstrate specific devices, or techniques, to the student. Clearly, such experiments remain a necessity in the early years of the course, in order to provide the fundamental understanding of the subject which is needed before design skills can be acquired.

Practical work in the final year takes the form of an extensive, open-ended project in which students, with minimum supervision, are encouraged to use appropriate technology to solve original design or research problems. In recent years the techniques used in final-year projects have escalated in complexity so that, for example, original solutions involving purpose-designed microprocessor hardware have become commonplace.

In this context, the third-year laboratory was seen as a transition from the closely-supervised traditional experiments of the second year, to original design work in the final year. To be more specific, the third-year laboratory must incorporate two important features:

(i) experiments involving the new microelectronics technology and
(ii) open-ended experiments requiring greater initiative on the part of the student.

3 THE NEW LABORATORY

3.1 *Organisation*

At the outset, it was felt that the six hours traditionally allotted to each experiment in the first three years' teaching laboratories would be insufficient for the 'project'-like experiments envisaged. This, combined with the limited physical space available, lead us to propose that students spend alternate weeks in the laboratory, working for up to twelve hours on single experiments. With a total of twenty working weeks in the Autumn and Spring terms, ten experiments were then needed. In addition, single student work stations were to be the norm, students pairing up only where unusually expensive equipment was involved.

In keeping with the project nature of the new laboratory, it was felt that intensive supervision would be unnecessary, indeed, possibly harmful. A reasonable amount of supervision should, nevertheless, be available if required, at the start of each week. Accordingly, the fifteen-work-station laboratory is manned by three staff, and one postgraduate demonstrator for the first afternoon session (of three hours), and then, for the remaining three afternoon sessions, just one member of staff, and one demonstrator. A laboratory technician is also present during the whole twelve-hour period, to deal with any problems with the equipment.

3.2 *The experiments*

An important consideration, when devising actual experiments, was that they should be multi-layered, so that there is sufficient material for the weak student to become involved and learn at his own level. At the same time, additional ideas are available for further exploration by the more able students.

Appendix 1 contains extracts of the actual laboratory handout sheets for those experiments which, in the authors opinion, best represent the philosophy of the new laboratory.

The following is a breakdown of the subject areas covered in the laboratory, and details of the experiments within each area.

3.2.1 *Analogue computation* Despite the advent of powerful digital computers, analogue techniques still provide the quickest solution to certain types of problem. Indeed, there has been a recent resurgence of interest in hybrid computational techniques for the control of complex structures such as robots. This experiment introduces students to the principles and practice of analogue computation via a series of graded exercises leading up to the design and implementation of a patch for a problem requiring simple iterative and hybrid techniques. Able students are encouraged to devise and patch problems allied to their own particular fields of interest rather than being rigidly guided by the laboratory sheet.

FIG. 1 The teaching laboratory.

3.2.2 *Communications* The degree course, in this department, has a significant communications component throughout. Accordingly, we felt justified in including two experiments to investigate advanced communications techniques, one in analogue communications and another in digital communications, both to use state-of-the-art devices and techniques.

The analogue experiment is based upon a full specification SSB transmitter module, available commercially, and intended primarily for amateur application. The experimental transmitter modules have been assembled such that all of the internal signals are accessible for examination. Integral with the experiment is a sophisticated spectrum analyser and the primary objectives of the experiment are twofold; to evaluate the performance of the transmitter module — and verify (or otherwise) the manufacturer's specification — and, at the same time, become familiar with all of the techniques of spectral analysis.

The digital communications experiment is designed to demonstrate the techniques and problems of Pulse Code Modulation (PCM). The experimental hardware consists of four functional modules; an analogue to digital converter, a digital to analogue converter, and two universal asynchronous receiver transmitter ICs (UARTs), set up in parallel to serial, and serial to parallel configurations. The modules may be operated singly, or in combination, the student having to design the interconnection, timing and filtering in order to construct a model serial PCM system.

3.2.3 *Design and breadboard* It was felt that students in the third year should rapidly acquire the basic skills of design and breadboarding and the ability to work from data sheets and specifications. In addition, we wanted to introduce them to the discipline of getting components from our stores system and negotiating with storekeepers. To this end, two design experiments were evolved, one predominantly digital, the other analogue.

The digital experiment consists of a design specification for an electronic combination lock safe. The objectives are to design and breadboard the system to the working prototype stage. Students are given data sheets on a keyless-lock IC and a hexadecimal keyboard as a starting 'hint'. From then on they can draw any components they need to build up the system. The Radio Spares catalogue is also available, and students are expected to consult data books and data sheets in the departmental library as needed. In this way, the organisational and practical problems of getting a design from paper to breadboard are rapidly brought home.

The analogue experiment is centred around a number of analogue integrated circuits: the 3900 Norton amplifier, 555 timer, and phase lock loop, frequency to voltage converters etc. Students are given the data sheets on these devices and asked to investigate some of the suggested applications shown. An objective of the experiment is to design a 'system' using analog ICs, and an example is suggested which is a gas sensor alarm — telemetry system. Students are, however, encouraged to think of new applications and design and build their own system.

3.2.4 *Microprocessor interfacing* This experiment aims to teach the student the basic techniques of interfacing a microcomputer to the real world, something that is often neglected in 'traditional' microprocessor courses. Each workstation has an Apple II microcomputer, with disk drive, and a variety of peripherals are available. The advantage of using a sophisticated microcomputer as the controller, as opposed to a lower-level microprocessor development system, is that the student can work in both a high-level language and a machine code environment, and quickly develop powerful control algorithms. The peripherals that are available include a Smart-Arms robot, a general purpose input-output module with push button switches, indicating LEDs, and analogue potentiometers, and a set of commercial interfaces manufactured by Feedback. These include a simulated washing machine, a simulated traffic lights/pedestrian crossing system, a heating control system, and a stepper motor module. Students are presented with documentation on the Apple and peripherals and are expected to choose what they wish to do. The excellent documentation supplied with the Apple II enables weaker students to quickly brush up their BASIC with a 'tutorial', whilst stronger students are expected to design their own interfaces and peripherals. To date, two such interfaces have been designed by students on their own initiative, a light pen, and a graph plotter interface.

3.2.5 *Microwaves* The unusual nature of microwave hardware and devices means that students rarely have an opportunity to experiment first-hand. This single experiment is designed primarily to provide this opportunity, and is based upon the Sivers microwave components — designed particularly for educational use. The hardware consists of a Klystron as a microwave source, and a selection of microwave devices and waveguides which may be bolted together to form a complete working system. The student is asked simply to become familiar with the operation and use of each device, and finally verify a number of fundamental relationships, for example, the relationship between free-space and waveguide wavelength.

3.2.6 *Software* The laboratory includes two experiments based on the production of computer programs. This reflects the authors' view of the increasing importance of software techniques in electronic engineering. The first experiment is documented in Appendix 1.1 and aims to encourage students to produce not just a working program, but a fully-documented piece of software of an industrially acceptable standard.

Computer simulation is a technique used increasingly by the electronic engineer as a method of design verification, before any effort is committed to actual hardware. The second software-oriented experiment is designed to demonstrate the use of computer simulation. The student is asked to model mathematically a circuit, or system, either real or imaginary, and produce from this model a working simulation capable of verifying, or otherwise, the original system specification. The key requirements of this exercise, are that the model is an accurate representation of the specified system, and that the final program truly simulates the model.

3.2.7 *Systems analysis and control* This experiment seeks to introduce the student to sophisticated network analysis equipment and also to provide a practical basis for work covered in the third-year Automatic Control course. It is in two parts each of which is expected to occupy about six hours. The first part allows the student to gain familiarity with programmable network analysis equipment (the system comprises a Hewlett-Packard desk-top computer controlling a frequency synthesiser and gain-phase meter) and to use frequency response techniques to identify an unknown linear system. The second part requires the modelling and control of an analogue system (a d.c. servo) and a comparison between theoretical and measured behaviour. During the course of the experiment, students are encouraged to make full use of a set of interactive control system design programs available on the departmental PDP11/34 computer and thus gain further insight into the role of computer-aided design in modern engineering practice.

4 ASSESSMENT

Assessment in the new laboratory takes the form of formal laboratory reports submitted for each experiment. In addition, students are required to keep day books as personal records of work done. Formal reports are written in the alternate 'week-off' between laboratories.

Whilst this is not itself a departure from standard practice, the actual format of formal reports required has been radically altered (Appendix 2). In particular, it is felt that the formal report should not follow a standard pattern but should more closely reflect the actual nature and purpose of the work carried out. For example, in the case of the 'Production of technical software' (Appendix 1.1), the formal report comprises a fully-commented listing and users' and programmers' guides.

5 CONCLUSION

The new laboratory is now approaching the end of its first year of implementation, and appears to have been a major success. We have noted a greater response from students, in developing their own ideas within the framework of the experiments, from the whole range of abilities within the class. In particular the 'layering' of experiments seems to work well, in that the weakest students do not experience difficulties, whilst the more able students do not lose interest, as is often the case with traditional tightly-controlled experiments.

REFERENCES

[1] The Finniston Report: *Engineering Our Future, Report of the Committee of Inquiry into the Engineering Profession, Command Paper No. 7794,* Her Majesty's Stationary Office, London (1980).

APPENDIX 1.1

Experiment 1 Production of technical software

Objectives The aim of this experiment is to write and document to an industrially acceptable standard a piece of software to solve an engineering problem.

The Problem A large number of different problems have been selected and students will be allocated one of these at random at the beginning of the experiment. However, any student wishing to tackle a problem of his own devising is encouraged to do so, provided he first discusses the project with Dr. G. E. Taylor.

The Programme Choice of level and type of language and of machine is left to the student, but some justification of this choice should be made in the Programmers' Guide (see section on documentation). *Among* the factors to be considered are

(i) portability of high level languages such as FORTRAN, PASCAL and ALGOL
(ii) speed of low level implementations
(iii) general, multi-user accessibility of main frames
(iv) usefulness of programs to run on personal computers since these are now widespread
(v) desirability of graphical output
(vi) machines available! — any university or departmental computer normally available to undergraduates may be used, alternatively a student may use his own machine, but see note on assessment.

Programs for programmable calculators are acceptable provided that the writer demonstrates the usefulness of solving the given problem on such a machine.

Documentation The source file itself should be *fully* commented unless this is impossible as with a program for a programmable calculator in which case a list of codes plus comments should be supplied. In addition two pieces of documentation are required.
(1) A Users' Guide written in such a way as to enable an engineer who is not conversant with computers to understand the task performed by the program, to enter his own data successfully, run the program and interpret the output.
(2) A Programmers' Guide detailing *briefly* and *clearly* the algorithm, data structure and so on to enable another programmer to understand and amend the software.

Assessment A standard formal report is not required. The work will be assessed from an examination of the Users' and Programmers' Guides and the commented source together with a demonstration run of the program.

APPENDIX 1.2

Experiment 5 — A P.C.M. link

1 *Objectives and Overview* The key elements of any communication system employing Pulse Code Modulation, are the analogue to digital conversion (A–D), and digital to analogue conversion (D–A), the modulation and demodulation processes respectively.

Practical A–D and D–A convertors usually have parallel digital interfaces, and so the additional processes of parallel to serial, and serial to parallel conversion are incorporated to give a serial pulse coded bit stream.

This experiment models PCM modulation and demodulation using off the shelf MSI components; the ZN425E 8 bit A–D/D–A converter, and the 6402 Universal Asynchronous Receiver Transmitter (UART). Two of each of these devices are supplied, so that the individual devices may be evaluated in each of their dual modes — and ultimately all four may be interconnected, to model a complete PCM link.

The objectives of the experiment are:
(a) To gain familiarity with these devices.
(b) To evaluate the devices over the complete range of operational parameters.
(c) To appreciate the problems, and concepts of Pulse Code Modulation.
(d) To model a PCM system.

2 *Experimental* A suggested programme of experimentation is as follows:
(a) *A–D, D–A Converters*
Test the converters separately using DC input voltages, and binary switches. Use the Farnell pulse generator to generate single TTL level pulses, to 'single step' the A–D converter.

Calibrate and test for linearity.

Link the digital output of the A–D to the input of the D–A, and compare analogue input of the A–D, with analogue output of the D–A.

Clock the coupled A–D/D–A over a range of speeds, and analogue waveforms. See the enclosed diagram showing recommended signal generators.

Define the useful ranges of clock, conversion and analogue frequencies.

Observe the effects of 'stuck at' bits in the digital interface, and the effects of exceeding the analogue frequency limits.

Design and build an output filter for the D–A converter.
(b) *UARTs*
Single step the UARTs. Observe the serial output for different parity, length and stop bit

options. Generate a serial bit stream manually, and clock this into the serial to parallel UART.
Transfer an 8 bit word across the interface by manual clocking.
Observe the effect of mismatching the parity, length and stop bit options.
Clock the UART pair at high speed, and observe the bit stream using an oscilloscope and logic analyser combination.

(c) *The PCM link*
Connect, using the ribbon cables, the units to form a serial PCM link. Run the link over a range of speeds, and analogue input waveforms (including speech), observing the digital data using the logic analyser.

Notes: Please ask a demonstrator to check your power connections to A–D/D–A and UART boxes before you switch on. *Ensure* that signal generator outputs used for clocking and control are TTL levels (0–5V).

3 *Assessment* The formal report should be an assessment of suitability, for application in an actual PCM communication system, of the ZN425E and 6402 devices. Include a design for a control circuit, to provide all of the timing and control signals, for the system.
Inclusions: Circuit diagrams and data sheets.

APPENDIX 1.3

Experiment 8 Apple interfacing

Introduction This experiment is all about controlling peripherals with the Apple II microcomputer. You are provided with a number of interfaces to the Apple, and a number of peripherals. These include:

(1) Buffered version of the Apple's Games/I/O Connector.
(2) Simulated washing machine.
(3) Simulated traffic lights.
(4) Heat control system.
(5) Stepper motor.
(6) Smartarm robot.

You will have to make full use of the Apple and peripherals documentation provided.

What you are going to do You are going to control the various peripherals by writing control programs in BASIC.

How to go about it

(1) You must be able to program in BASIC. If not, learn quickly by reading the APPLESOFT TUTORIAL.

(2) Learn how you can make I/O lines go 'high' or 'low' by use of the POKE command. Also how you can sense the state of an input line by use of the PEEK command. Use the Buffered games I/O connector, turn output LEDs on and off, sense the state of the input lines and the angle of the pots. Use the Applesoft Manual and the Apple Reference Manual.

(3) Choose one of the FEEDBACK peripherals, read the manual and write a BASIC program to control it.

(4) Control the Smartarms Robot. Get it to do a simple pick and place task. Then more complicated tasks if you wish. Remember you must share the robot.

(5) *Either* (a) Try some of the other FEEDBACK peripherals, *Or* (b) Design and breadboard your own peripheral, interface it to the Apple via the buffered games I/O connector, and control it from a BASIC program.

Hints:

(1) Keep your programs simple and well documented.

(2) You can save a program by typing SAVE filename. Load a program by typing LOAD filename. Don't get bogged down in the Apple's DOS.

(3) Double check power connections.
(4) Don't blow up the Apple.
(5) You can take the documentation away during your week 'on', and the subsequent report week. If you lose it we will charge you.
(6) Don't be afraid to do your own thing in this experiment — but discuss it with a demonstrator first.

Report Requirements
(1) Your report should tell me what you did and what the results were.
(2) Don't tell me about the Apple, peripherals, robot. I know more than you about these and don't want to read chunks of regurgitated manual.
(3) Include listings of your programs, and circuit diagrams of your *own* peripherals.
(4) Keep it short.

APPENDIX 2

Guide to writing formal laboratory reports

1 *Overview* The preferred style and format of laboratory reports in the third year lab is that of a technical report. This will be a significant departure from what you have become used to in previous years, and will approach the type of report required in the real engineering world. Imagine the laboratory experiment as an evaluation of a new equipment, or technique or concept — and you are then to report the results of your evaluation (to a project manager, for example). In a real situation you would be testing out the equipment for possible application to an engineering problem — and on the basis of your technical report the project manager would make the decision of whether to incorporate the equipment or not. The technical report should then be:

(i) Brief. Your time, and your project managers' time is valuable.
(ii) Accurate. Your project manager is relying on *your* report to make his decision (which could be expensive) — so get your facts right.
(iii) Descriptive rather than mathematical. If you have to quote mathematical results include a verbal description explaining what the mathematics *really means* in engineering terms. Your project manager may not be an expert in the field, that's why he employed you to do the evaluation! If you must include proofs put them in an appendix.
(iv) Informative. Don't tell your project manager what he knows already — and don't include a blow by blow account of how you undertook your evaluation, he's not interested!
(v) Conclusive. Don't leave your report open ended — otherwise you'll be asked to write another.

2 *Format* The main body of your report should not be more than 5 sides of A4, excluding graphs, diagrams, appendices and references. Reports significantly longer than this will be penalised.

The actual format will of course depend on the nature of the experiment, but will probably consist of:

(i) *An Abstract* This, *very* briefly states what the report is about — the object of the work, and major conclusions of the report. (Most people decide whether to read a paper or not, according to the abstract).
(ii) *Introduction/overview* It is difficult to generalise upon this/these sections, because this part will depend very much upon the actual work, but here you will 'fill in the background' needed to appreciate section (iii). (i.e. a theoretical summary).
(iii) *Results summary and discussion* This is the most important part of the document — and contains the actual findings of the work — with reference to results (experimental or theoretical). The results are examined from an engineering viewpoint.

(iv) *Conclusions* Using the previous analogy the conclusions would make *actual* recommendations. (i.e. is the technique useful/applicable/relevant/reliable etc?).
(v) *References*
(vi) *Appendices* (where applicable). Put detailed results, and working in here, if you feel it must be included.

3 *Presentation* Almost any presentation is acceptable (except loose sheets), provided it is *legible* without the assistance of a handwriting analyst!

4 *Submission commitment* A Report is required for *every* experiment, and will be collected by the laboratory technician on the first day of the subsequent experiment. (Every two weeks).
Reports one day late will probably suffer a penalty, any later and you risk zero marks.

5 *Special requirements* Most of the experiment sheets will contain a note with specific requirements for the formal report.

ABSTRACTS–ENGLISH, FRENCH, GERMAN, SPANISH

An advanced electronics teaching laboratory
The rapid advancement of electronics technology means that undergraduate teaching laboratories should be frequently reviewed. This philosophy led to the new third-year laboratory structure described in this paper. This was implemented in October 1981 as an integral part of a four-year course and, in a radical departure from standard practice, involves open-ended experiments including both software and hardware design.

Un laboratoire d'enseignement d'électronique de pointe
Le développement rapide de la technologie électronique entraine une fréquente remise à jour des laboratoires d'enseignement a l'usage des candidats ingénieurs. Cette philosophie a conduit à la nouvelle structure du laboratoire de 3 ème année décrit dans cet article. Il a été réalisé en octobre 1981 comme partie d'un cours s'étendant sur quatre ans, et, en contradiction totale avec la pratique courante, implique des expériences sans limitations, utilisant à la fois la conception de logiciel et de matériel.

Ein fortgeschrittenes Unterrichtslaboratorium der Elektronik
Der schnelle Fortschritt der Elektronentechnik erfordert, dass Unterrichtslaboratorien für Studenten häufig einer Revision unterzogen werden müssen. Diese Philosophie führte zu der neuen, in dieser Arbeit beschriebenen Struktur des drittjährigen Labors. Diese wurde im Oktober 1981 als ein wesentlicher Bestandteil eines vierjährigen Kursus eingeführt; radikal von Normalmethoden abweichend ist dies mit Versuchen ohne vorbestimmte Ergebniße, sowohl beim Software-, wie beim Hardwareentwurf, verbunden.

Un laboratorio para la enseñanza de electrónica avanzada
El rápido avance de la tecnología electrónica conlleva el que los laboratorios de enseñanza para no graduados deban ser actualizados con frecuencia. Esta filosofía lleva a la nueva estructura de laboratorio de tercer curso descrita en este artículo. Se puso en marcha en octubre de 1981 como parte integrante de un curso de cuatro años y, dentro de un alejamiento radical de la práctica habitual, aborda experimentos abiertos incluyendo diseño tanto del software como del hardware.