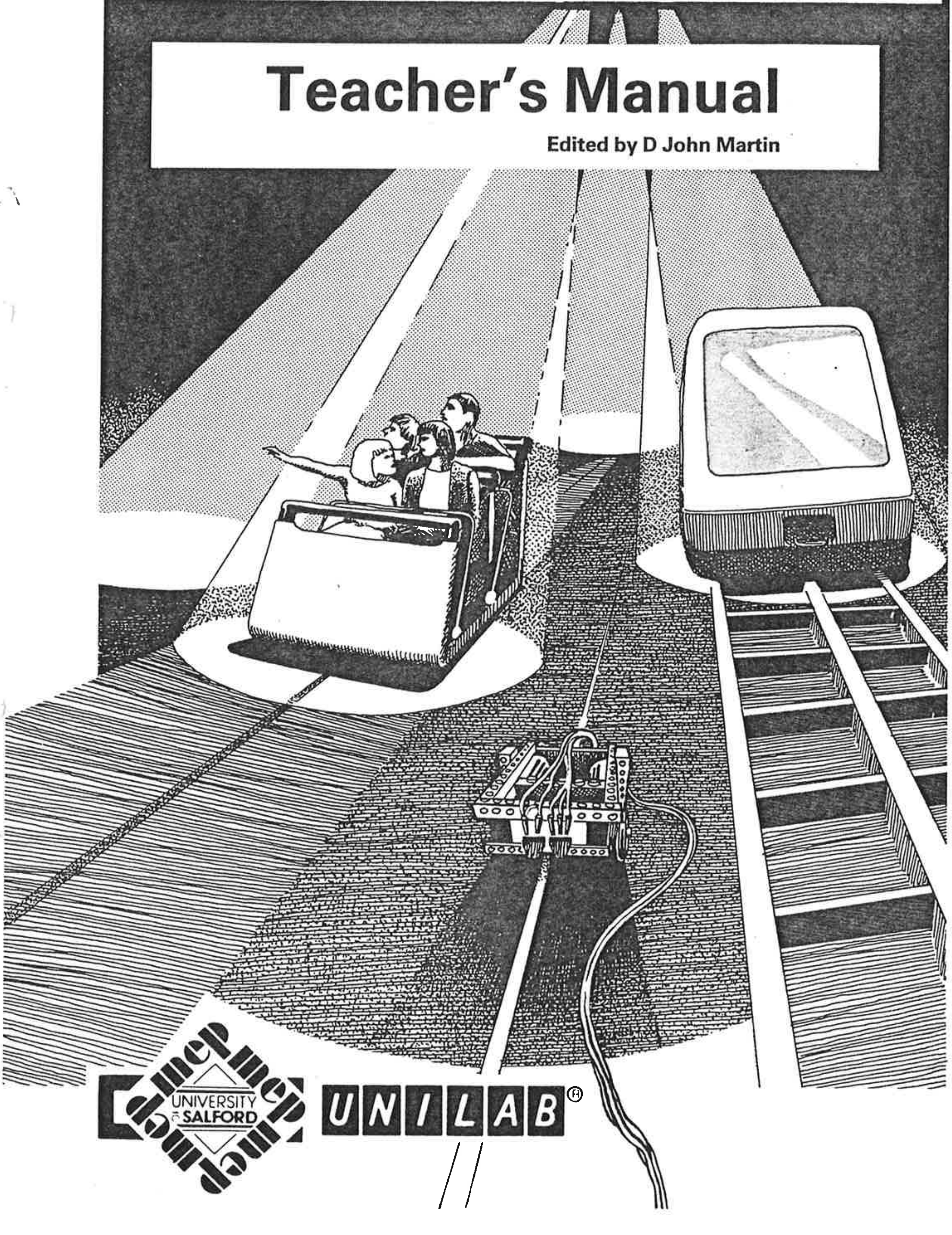
Control Pathways

Teacher's Manual

Edited by D John Martin



University of Salford

UNILAB®

Contents

"In training a child to activity of thought, above all things we must beware of what I will call "inert ideas" - that is to say, ideas that are merely received into the mind without being utilized, or tested, or thrown into fresh combination."

A. N. Whitehead (in The Aims of Education)

"Life is a game.

In order to have a game
something has to be more important
than something else.

If what already is,
is more important than what isn't,
the game is over.

So, life is a game in which what isn't,
is more important than what is."

Werner Erhard.

I am grateful to A C Black Ltd for permission to use the copyright quotation from A N Whitehead and to Werner Erhard and associates for permission to use the copyright (1973) quotation from "Aphorisms" by Werner Erhard.

1. Introduction

Control Pathways is a range of resources designed to support a practical-based introduction to one of the most important applications of microprocessors - as the heart of programmable control systems.

At present many schools offer awareness and examination courses in computers and also use computers for computer-based learning and information retrieval and processing. In recent years work with electronics in schools has also increased and many schools offer courses that give pupils an insight into how the basic building blocks of modern electronics - sensors, gates, counters, amplifiers, etc. - are used.

Control Pathways is aimed to help schools fill an important gap in this range of provision. Modern computer courses stress the input - process - output overview of the system. But the input device used is almost invariably the keyboard and the output device the screen. In consequence the range of applications explored is restricted. Modern systems-based electronics courses lay a similar stress on input - process - output (with a range of input and output systems) but the processing is usually performed by non-programmable building blocks. In consequence the range of control problems encountered is unrealistically narrowed.

Most microprocessors in the "real" world are used as the programmable processor in electronic control systems - from washing machines to robots. And most modern electronic control systems include a microprocessor.

A balanced introduction to computer systems will therefore include work on control and a balanced introduction to electronics will include work on the microprocessor as a "processing" block.

There is a second (and in my view more important) reason why an introduction to work with microprocessors for control is valuable. Control is one of the most exciting and interesting problem-solving areas available to young people. The solution of a typical control problem involves:- thinking about the problem as a whole, selecting suitable input and output devices, (often) the design and contruction of a mechanism, describing the control problem in a clear logical way, interpreting this information, converting generalised statements into specific instructions - with the "pay-off" of developing a system that performs in a way that was planned.

At its simplest level it can simply involve the satisfaction of following instructions to produce something that works to perform an interesting task.

"Control Pathways" is a structured introduction to the concepts of control. The control of most systems (and not just electronic ones) embodies certain very fundamental principles; it is these that this learning material aims to address through the medium of programmable microelectronics. Electronics because more and more control is handled electronically as time goes by, electronics because it's cheap, easy, flexible and versatile - the reasons why industry uses electronics to control.

Each of the sections of the main pupil booklet introduces a new concept, with the most essential, and where possible the most straightforward, concepts met first. As far as possible, real-world examples are used and models of such control systems built and programmed. Details of the "conceptual route" are as follows:

Section 1 deals with outputting control information - you can't do anything unless you can switch parts of the external system on and off.

Section 2 shows how to implement time delays. This makes possible the **sequence**, perhaps the most basic of all "control" programs.

So far the pupil has turned fully on or fully off the control voltage on individual control lines from the controller. This enables a lamp or a motor to be switched on or off, a motor to be reversed, or a control code to be output to the Music Module to sound a particular note. **Section 3** shows how the **brightness** of a lamp or the **speed** of the motor may be controlled. This requires a control code (which might represent a number on a scale from 0 - 7) to be output to an external "black box" that can control the voltage applied to the lamp accordingly. The concept of Digital to Analogue Conversion has been introduced (and thus the concept of how a variable or "analogue" quantity may be controlled by a "digital", or on/off, controller).

Some control systems function purely as sequencers - the outputs are a function of time only - but most systems control outputs in response to changing **input** conditions. **Section 4** shows how to input information to the controller, and how to design the program so that the input states affect the control of the outputs.

Almost all real-world control systems employ **feedback.** This means that information about (for example) the position of an arm is fed back to the input of the control system. The controller is then able to "drive" the arm more intelligently, basing its control decisions on where it knows the arm actually **is.** Without feedback, the controller must attempt to calculate or remember where the arm **ought** to be. The controller will not know if the arm moves more slowly than expected, hits an obstacle, or reaches the limit of its travel. **Section 5** builds on the principles of outputting and inputting and introduces this essential concept of feedback. At this stage the feedback is only two-state, or yes/no; feedback in a system controlling and monitoring analogue quantities is reserved for section 7.

Section 3 allowed pupils to see multi-level, or analogue, outputs. **Section 6** illustrates analogue **inputs.** It is often necessary for a control system to "read in" an analogue quantity, for example a temperature, a light level, or the level of a liquid in a tank. The controller itself can only "understand" on/off, or digital information, so this time an Analogue to Digital Converter is needed. The other important concept introduced in this section is that of **processing** information electronically.

Section 7 takes over the subject of feedback where section 5 left off. Pupils now develop control systems which sense analogue (multi-level) quantities and control outputs proportionally.

Section 8 presents a wide range of ideas on **project possibilities.**

As well as the gradation in difficulty through the sections, there is a gradation through any individual section. Each section contains a number of "extension" modules with a substantial design ingredient. Only the most able pupils will finish all the work in a section. Depending on the time available for the work, it is envisaged that most pupils would typically tackle only a few of the extension exercises in a section. At the same time, the "core" modules aim to be very supportive, with full software listings for the chosen controller given in the mini-manuals. The first few core exercises in sections 1-6 should be within the reach of almost all young teenagers, therefore. Later exercises within a section are mainly for reinforcement, and for the development of an idea. Slow pupils will get more from an overview gained from tackling a couple of exercises in each of sections 1-6, than from using the available time to plough though every exercise in sections 1 and 2. Their interest is also more likely to be sustained in this way. In other words, teachers will need to use their professional judgment in choosing the right moment to encourage an individual pupil to move on to the next section.

Control Pathways was very much a team effort. It was first conceived in 1982 when Peter Nicholls and the editor realised that Peter's 3-Chip Plus microprocessor controller system (which was funded by MEP) and MEP Microelectronics for All (MFA) could be designed to control the **same** set of devices. We therefore arranged that the Memory Module of MFA and the Computer Module of MFA could drive the same outputs and accept the same inputs as 3-Chip Plus. Two existing MFA Output Devices - the Movement Module (which can control a LEGO® buggy) and the Music Module - are existing MFA devices designed to suit **both** systems.

Over a twelve month period a team of teachers have worked to develop ideas for pupil activities and write the materials (including the materials in this Teachers' Manual. The team was:-

Jim Cunningham (Blackburn College, Lancashire)
Alan Harper (Nether Stowe Comprehensive School, Staffordshire)
J Oliver Linton (Harrogate Grammar School)
D John Martin (MEP Electronics Education Development Unit, Salford University)
Peter Nicholls (MEP Regional Director, East Midlands)
Richard Patterson (Bishop Challenor RC High School for Boys)

The materials, because they are new to schools, have no obvious "home". We have not designed them for a particular exam syllabus but rather to be a coherent and exciting range of resources, accessible to pupils from about age 13 up to 6th form level (depending on background). The materials can be used to support a range of examination and non-examined work and are sufficiently flexible for the route chosen to emphasise particular aspects.

Before encountering Control Pathways pupils should have already encountered work on electronics systems. While MFA is a natural lead in (in view of the hardware compatibility) it is not a pre-requisite and a practical-based introduction to electronics in which pupils encountered binary signals and patterns and used these concepts for control would form a suitable basis.

Likewise some practical-based awareness of programmable systems would be useful (but not essential).

Some of the extension exercises involve work with mechanisms built from kit systems (LEGO® , Fischer Technik® or Meccano®). These can be treated in various ways e.g. provided ready-built. However, if the pupils are to construct them themselves they will need some previous introduction to this (such as would typically occur for example in CDT:Technology).

One unusual feature of Control Pathways is that the main pupil booklet does **not** refer to a particular controller. The main booklet suggests control problems and gives general guidance. Anything which is specific to a particular controller is in the relevant mini-manual for that controller.

The initial reason for this approach was to give schools considerable freedom to choose what they would use as a controller. As well as the obvious economic advantage this means that the central concepts of control - which are **independent** of the controller **and its language** - are all in the main pupil booklet. These represent the core concepts for control and it is useful for pupils and teachers to see these isolated from what sometimes looms large - the controller and learning how to program it.

What matters **educationally** is the **process** of tackling the varied control problems and appreciating the relationship of the problem to the sensors, the output devices and a clear statement of the control requirement - the "algorithm". The passionate debates on 6502 or Z80 or BASIC or Logo are arguments about whether to use a pencil or a biro. The important skill in literacy is writing as a contribution to interpersonal communication. In control the important skills are logical (and sometimes lateral!) thinking as an expression of capability in serving human needs and human fulfilment.

I am very conscious that we are breaking a lot of new ground with these materials. A great deal of effort has been made by the writing team and by UNILAB to try to eliminate all possible misunderstandings or errors, but in such a wide range of materials it is almost inevitable that something will get through the net. UNILAB have agreed to a second edition of these resources if any such problems are encountered so please let me know of any improvements you consider should be included.

D. John Martin.

D John Martin
MEP Electronics Education Development Unit
Salford University
Salford
Lancs M5 4WT

2. Educational Aims

"Control Pathways" was developed to provide an attractive, exciting and motivating introduction to programmable control systems and their applications. The materials aim to foster awareness, understanding, capability and expertise in an area of technology of central importance which requires (for its full implementation) investigation, planning, design, making and evaluation applied to mechanisms, electronic input/output systems and software.

"Control Pathways" is designed for pupils of a wide range of abilities. Each section of the main booklet introduces a new concept.

In the first **core** exercise of each section the full solution to a control problem involving this concept is given. These core exercises are designed to be accessible to the **full ability range.**

In addition, the illustrative material used and activities suggested in this manual in section 4, aim to support **awareness** of economic and social implications of the technology. In further **core** exercises of each section other problems involving the same basic concept in different control situations are presented; a full solution is again given but pupils are encouraged to demonstate and reinforce their understanding of the solution by **modifying** the given solution to achieve a modified control requirement. These modifications are designed to be accessible to the **large majority** of the ability range.

In the **extension** sub-sections (which follow the core sub-sections) control situations involving the key concept of the section are presented. Only partial solutions are given and pupils are invited to apply their knowledge of the concepts and techniques introduced to achieve the required control.

There is considerable variation in difficulty between the various extensions; they are designed so that most pupils can tackle at least some extensions and those pupils who show ability in this type of problem-solving work can tackle challenging assignments. The full process of design, implementation and evaluation of a programmable control system is presented in simple terms to pupils (sub-sections 2.7 of the main pupil booklet). It can be summarized as:-

1. Information and overall planning
2. Select suitable input and output systems
3. Describe specific control requirements
4. Flow/Structure Diagram
5. Program
6. Test separate aspects and then test complete system
7. Correct if necessary

The basic philosophy of "Control Pathways" is to present a wide range of control problems and to gradually require pupils to carry out more and more of these stages for themselves. The booklet ends with project possibilities which require mastery of the full process for their effective completion.

The materials are therefore designed so that aims listed under Sections 1 to 5 below are accessible for all; the more extended aims are accessible to most.

The detailed aims of ''Control Pathways'' are:-

1) Motivation

(a) To provide a stimulating medium in which pupils can develop ideas of their own and explore the consequences of their own decisions.

(b) To give pupils confidence and the experience of success by providing them with a wide variety of realistic problems which have solutions at many levels.

2. Communication

To encourage communication by a variety of means including the use of clearly and logically expressed spoken and written English, flow charts/structured flow diagrams, to clarify the design of and to describe the solution to control problems and as an aid to testing and evaluation.

3. Co-operation

To foster attitudes of co-operation and participation by encouraging collaboration in small-group problem solving.

4. Numeracy

To reinforce and consolidate manipulation of numbers in the context of a varied series of applications.

5. Awareness

(a) To foster awareness of the varied ways in which programmable control systems are used in everyday life.

(b) To encourage awareness of social and economic implications of microprocessor control systems.

(c) To ''demystify'' the microprocessor and promote awareness of its dependence on human design and testing.

6. Understanding

(a) To develop appreciation of the concepts of **digital** quantities, **analogue** quantities, **input**, **process** (by a flexible **programmable** controller), **output** and **feedback** in control systems.

(b) To promote recognition of any programmable control system as producing:

(i) in all cases, an output pattern, or sequence of output patterns, in response to **stored instructions.**
(ii) in most cases an output pattern, or sequence of output patterns, in response to stored instructions **and** an **input pattern.**

(c) To encourage an appreciation of the **relationships between** the control requirement, the system output, the system input and the stored instructions.

(d) To provide practical experience and functional working knowledge of a range of input and output systems.

(e) To stimulate the exercise of judgement and discrimination in the selection of input and output systems appropriate for a particular control requirement.

(f) To provide a technical basis upon which informed consideration of the role of microprocessor control can take place and to encourage such consideration.

7. Capability and Expertise

(a) To stimulate a logical and systematic approach to problem solving.

(b) To foster precision and attention to detail in the solution of problems.

(c) To promote capability in the design, implementation and evaluation of control systems conceived and planned **as a whole** and involving as appropriate:-

(i) Mechanical systems (including design and constructional skills associated with these)

(ii) Input/Output systems

(iii) A programmable controller (including design of a suitable flow chart or structured flow diagram and its ''translation'' into a sequence of instructions.)

3. Classroom Management

I know that you are a busy teacher with lots to do. Please read this section. It will save you a lot of time.

1. The materials are planned to be flexible but certain basic assumptions have been made about methods of working. In particular it is assumed that pupils will work **at their own pace.**

Each section of the main pupil booklet introduces a new concept and the **core** exercises aim to claify and establish that concept. Therefore all pupils should do **all** core exercises **in sequence.** The **only** exception to this is in sub-sections 1.1 and 1.2 which are **alternatives.**

Having completed the core exercises (in which an initial solution is given but which typically involves some later modification by the pupils) most pupils should be able to tackle **extension** exercises in the section. It is **not** assumed that all pupils will do all extension exercises. Rather it is envisaged that pupils, in consultation with the teacher, will tackle one or two extension exercises per section. Different groups should be **encouraged** to tackle **different** extensions so that a rich variety of work will result in the class. Pupils' attention can be usefully drawn to interesting work of **other** pupil groups.

It is assumed that, working in this way, there will be quite a wide variety of rates of progress. In view of this (and to reduce costs) the recommended number of peripherals will **not allow** all pupils to do the same exercise simultaneously.

2. It is very difficult to estimate class time needed for this work. It will depend on what the pupils have encountered before, what controller is used, how many extension exercises pupils do, and whether all sections are tackled. It is estimated that, if pupils have worked with MFA or another system-based introduction to control and most tackle one or two extension exercises per section, plus a short project the time needed will be 20 - 30 hours.

3. Notwithstanding the above, it will probably be useful, particularly in the early stages of the work, to spend a few minutes clarifying some general points with the class as a whole or with part of the whole class group who are at about the same stage. For example - how to use the booklets, what to record, binary numbers and their conversion to decimal or hexadecimal, the concepts of input and output, the 8 input and output lines of the controllers, the way in which these lines relate to particular input or output devices, how to "read" a flow chart or structured flow diagram (you could take a non-electronic example - going through a door, baking a cake...or you can give a pupil a flow chart or structured flow diagrams of actions to carry out and have other pupils work out the flow chart/structured flow diagram being followed), the distinction (in a high level language) between a variable and its value or (in machine code) between an address and the value stored at that address.

If all the class is using the same, or similar, controllers (which is the simplest situation to manage) this can be extended to concepts associated with the controller e.g. how to connect the controller and load programs and particular language features.

As the work progresses and the spread of activities increase this will be less effective (and less useful). Instead it will become possible to explore the real world applications on a class basis. This might involve a short discussion of how a particular exercise relates to application the pupils know of, how **else** it could be done (this will need to take into account earlier work pupils have done on **other** control methods), what advantages there are to different approaches, are there any problems that might not be obvious from the exercise - safety, social aspects. Ideas for more extensive work are given in section 4 of this manual and teachers are strongly urged to plan to incorporate materials on these areas, particularly in the latter part of the work.

4. The pupils could work individually but, partly for economic reasons but mainly for educational one, the ideal working groups would be **pairs**. Indeed it may be that for many pupils the most valuable aspect of the work will be their discussion with their partner on tackling the problems. It is necessary to try to avoid one of the group becoming a "passenger"; this does tend to happen with pupils working in threes.

5. The pupils need access to the "Control Pathways" main booklet and the mini-manual for the controller they are using.

6. Each pupil must have an exercise book or file in which they will:-

(a) Stick in copies of the worksheets (see section 7 of this manual) and complete them

(b) Answer other questions (from the main pupil booklet)

(c) Develop solutions to problems

(d) **Most importantly**, build up their own table of the commands of their controller and how they are used. To solve the problems they **must** build up this table and will probably need encouraging to do this. Similarly they will probably need encouranging to tackle problems by **first** writing down a flow chart or structured flow diagram.

(e) You may also wish them to use the self-assessment sheets (see section 9 of this manual) which they should also stick in their exercise books. If they are using 3 Chip Plus they can also stick in coding sheets (see section 10 of this manual).

7. Some of the extension exercises involve the use of models built from kit systems. The reason for this was to provide a varied and realistic range of problems. A wide variety of approaches are possible:-

(a) Avoid these exercises.

(b) Use ready-constructed models. (This would require rather more kit systems.)

(c) Encourage pupils to tackle the constructions in permanent materials (wood, metal, plastic) as CDT projects and subsequently use the finalised models.

(d) Have some ready-made models and some the pupils build.

(e) Let the pupils build the models from kit systems. In this case more time will be needed and the pupils should have had some systematic introduction e.g. in CDT:Technology. Two pupil pairs could collaborate, one pair to build the mechanism the other to develop the electronic parts of the control system. Most models can be made from LEGO® (Technical Functions 1 and 2) or Fischer Technik® (computing kit) or Meccano® . The only exception is LEGO® which does not seem suitable for extended structures such as the gantry crane.

8. The hardware modules need to be stored separately and clearly labelled.

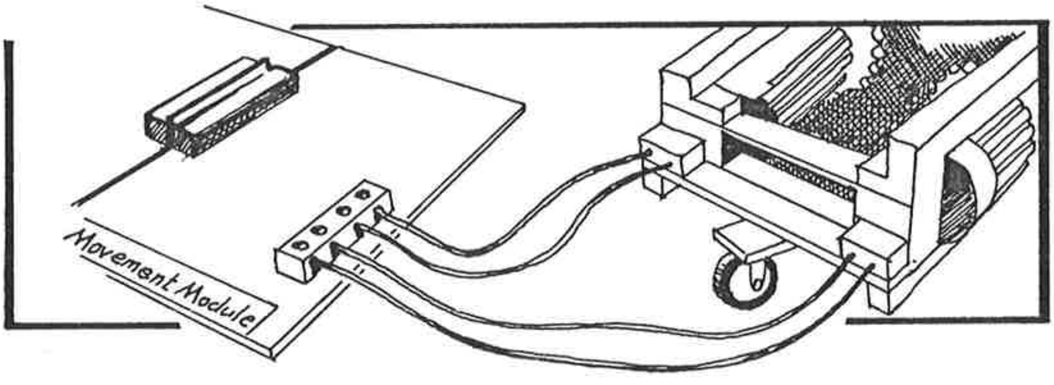
9. Pupils should be encouraged to return equipment to the correct place as soon as they have finished with it, ready for the next pupils. Otherwise pupils will start wandering around looking for modules.

10. Pupils should be encouraged to report faults or damage as soon as it is noticed.

11. Faults should be repaired as soon as possible.

12. Equipment should be checked at the end of each session.

13. Leave the connections to the buggy from the Movement Module intact (or colour code them). If the plugs are reversed the information in the pupils' main booklet on motor directions will not apply.



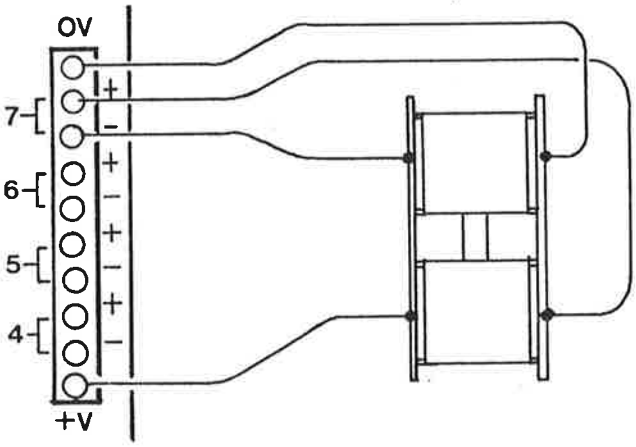
14. You will need a few odd bits and pieces:-

(a) A torch (sub-section 4.6)
(b) Insulating tape (sub-section 5.4 - for the line)
(c) Double-sided sticky-pads (to stick sensors on to models).
(d) Thermometer (sub-sections 6.3, 6.5, and 7.2)
(e) Small box (sub-section 7.2)
(f) Two 12V 24W bulbs (as used in cars) plus holder and (stackable) 4mm plugs
(g) Two 12V 6W bulbs plus holder and 4mm plugs
(h) Two 12 DC motors e.g. as used in Meccano® and 4mm plugs
(i) A stopwatch (if most pupils do not have watches) for sub-section 2.11
(j) At least four **stackable** 2mm plugs to connect sensors to the 8 Switch Input (needed for sub-section 4.4)
(k) A small plastic tub e.g. a margarine tub (for sub-section 4.7)

15. Particularly when working with a new output device, it is useful to put the 8 LED Output after the controller and before the device to see the binary pattern sent from the controller to the device.

16. The response of the analogue light sensor (which goes into the A to D Converter) is logarithmic. This was a deliberate design decision because of the very wide range of light levels encountered. However in darkness the sensor output does not go to zero as pupils might expect but (depending on the device characteristics) to 20 to 50.

17. The point motor is designed for use with Peco point motors and should be connected as shown:-



18. The pupil booklet ends with short ideas for projects. Fuller details are given in section 8 of this manual and pupils will need a copy of the information on their selected project.

19. When using the 3-Chip Plus controller it is suggested that, as an introduction to the work, a class demonstration with the "DEMO EPROM" will clarify some of the capabilities of the system. The EPROM stores a number of programs.

To demonstrate it (perhaps as an introductory class demonstration) insert the DEMO EPROM into 3 Chip Plus and plug the 8 Switch Unit into the input and the 8 LED Output into the output. There are 8 programs on the EPROM.

Put all switches, except the right hand one, down. The LEDs should flash on and off in blocks of four when RESET is pressed.

Put switch 0 back down and put up switch 1. Press RESET and watch the LEDs - you should now have a "chaser" display.

Just switch 2 gives a "chase" in the opposite direction. Only Switch 3 gives traffic lights.

Now plug the Music Module in place of the 8 LED Output. Switches 4, 5 and 6 give three different tunes.

Finally switch 7 up and reset gives a program in which the 8 Switch Input can be used to select the note being played.

20. After heavy use the gears on the LEGO® buggy may need lubrication with light oil so that the motors pull evenly.

21. If you use "System Alpha" in electronics the analogue blocks of that system can be used with the A to D Converter. The voltage from the Alpha module is reduced by a factor of 2 before feeding to the A to D Converter.

22. Details of the connections to the A to D Converter, the Sensor Module and the main connectors are given in Section 8 under "A Bolt-on Goody".

23. The following table shows the peripherals needed for each exercise in Control Pathways:-

	INPUTS					OUTPUTS											
Exercise	8 Switch Input	Sensor Module	A to D Converter	Multiplexer		8 LED Output	D to A Converter	Point Adapter	2 Digit Display	Relay Unit	Stepper Motor	Movement Module	Music Module				
1.1						*											
1.2												*					
1.3									*								
1.4										*							
2.1						*											
2.2						*											
2.3													*				
2.4													*				
2.5												*					
2 6						*											
2.8													*				
2.9									*								
2.10											*						
2.11										A		A					
3.2						*	*										
3.3							*										
3.4							*										
3.5							*										
4.1	*					*											
4.2	*											*					
4.3	*									*							
4.4	*									*							
4.5		*				*											
4.6		*							*								
4.7		*											*				
4.8	*					*											
4.9		*							A			A					
4.10		*											*				
4.11		*							*								
5.1		*								A		A					
5.2		*								A		A					
5.4		*								A		A					
5.7		*						*		*							
5.8	*									A		A					
5.9		*								A		A					
6.1			*														
6.2			*														
6.3			*			*											
6.4	*								*								
6.5			*						*								
6.6			*						A				A				
6.7			*			*											
7.1			*							A		A					
7.2			*							*							
7.3			*	*						A		A					

A = alternatives

Control Pathways

Projects	INPUTS				OUTPUTS								
	8 Switch Input	Sensor Module	A to D Converter	Multiplexer	8 LED Output	D to A Converter	Point Adapter	2 Digit Display	Relay Unit	Stepper Motor	Movement Module	Music Module	Output T Piece
Simon	*				*							*	
Sophisticated Doorbell		A										*	
Washing machine		*							*				
Security System		*			*				*				
Model railway		*					*		*				*
Robot arm		*							*			*	*
Paper tape organ		*							*				*
Central heating		*							*				
Puppet show		*							A		A		
Conveyer belt		*							*				
Pneumatic control		*							*				
Chart recorder/X-Y			*										
Communications		*							*				
Automated fish tank			*	*					*				
Coin dispenser	*								*				
Drilling Rig	*	*							*				
Complex traffic lights					*								
Bolt-on Goody													
Alarm for disabled	A	A							*				
Automatic greenhouse		A	A	*					*				
Solar panel tracker			*	*					A		A		
Learning robot			*	*					A		A		

A = alternatives

4. Social Aspects of Microprocessor Control

In public perception the microelectronics revolution is often associated with job loss - "replacing people with robots". Reality is more complex.

Sometimes automation has indeed reduced employment. Sometimes new employment and even new industry has been created. Often the impact of microelectronics has been in job change - with a need for re-training of employees.

It is strongly recommended that these issues are explored - perhaps in several ways - during the course. Here are some ideas on how this can be done:-

A visit to a firm and/or a talk

Very few schools are so remote that a firm of other institution cannot be visited which uses microprocessors in control and this is probably the best way of seeing the relationship of this material to the world of work. While industry is probably the major user of microelectronics for control, public service institutions e.g. hospitals, transport, make major use of this technology. To be valuable, it is necessary to find a firm where someone is willing and able to describe the use the firm makes of microelectronics for control and to **relate** this to the work the **pupils** are doing. It might be useful to discuss with pupils possible firms or other places using technology and for them to write to suitable people in these, describing the work they are doing.

Whatever method is used, it is important that the firm is given full information on the work the pupils are doing. Ideally, someone at the firm should first give a guided tour, pointing out applications and then give a talk and answer questions. The sorts of issues that it would be useful to ask him or her to discuss are:-

a) Why does the firm use microprocessors in control?

b) What sensors and output devices are used? (Often they will be as used in Control Pathways.)

c) What did it do before there were microprocessors?

d) How extensive is its usage of microprocessors?

e) What have been the benefits to the firm?

f) What problems were encountered?

g) What has been the effect on employees - job loss/job creation/job change?

h) What future plans does the firm have for this type of work?

Video

Video is a very useful resource for relating the class-based work to applications

BBC Schools Broadcasting are producing a series of three programs entitled "**Microelectronics**". These will first be shown in June 1986 and repeated in subsequent years. They cover concepts introduced by MEP "Microelectronics for All" but should also be useful with "Control Pathways".

Granada Television's "**Micro at Work**" is a series of ten 15 minute programs. In this time they provide varied illustrations of real life systems.

Programs 1 & 2 "Sorting things out". A sequence on sorting on the Tyne and Wear Metro

a) coin sorting in ticket vending machine

b) train sorting using track sensors

c) Freight marshalling yard at Masfien Hamburg.

Programs 6 & 7 "Robots in Action".

a) school-built buggy

b) industrial/military robots

(other programs show data processing and CAD applications)

BBC TV Science Topics Electronics.
This has a sequence with Jo Engelberger (of Unimation) on robot development. At the end of this there is a good application from Russell-Hobbs showing how their new jug-style kettle has microchip control with a temperature sensor and power relay. A contrast is made with the old method using a thermal cut-out.

The BBC Series "**Computers in Control**" contains quite a wide range of useful material.

The Video "**IT Takeover**" from Quest (available from the New Scientist) explores the use and abuses of "information" aspects of the technology.

Books

Suitable background reading for pupils are books in the Usbourne series e.g. "Robotics".

Suitable background reading for teachers are:-

"All Change" by P Dutton, P Nicholls and B Brest (NEC 1984)

"The Mighty Micro" by Dr C Evans (Gollancz 1979)

"Information Technology Revolution" by Smith (Longman, 1982)

"Automation and the Office Worker" (APEX, 1980)

"Player Piano" by K Vonnegut (Granada 1952)

"Shaping Tomorrow" (Methodist Church, 1981)

"The Silicon Factor" by Curnow and Curran (NEC, 1979)

"The Collapse of Work" by Jenkins and Sherman (Eyre/Methuen, 1979).

5. Control Methods

How are systems "controlled" in the world of industry and commerce? Control is necessary in most products of industry, and is also required in most of the processes that make the products. By no means all control is electronic (control systems can be mechanical, hydraulic, pneumatic, even biological) but electronics is taking over inexorably. There **must** be a reason.

Engineering is about fulfilling human needs at least cost, and the cost of a product is attributable to:
(a) the design cost
(b) the cost of the raw materials
(c) the cost of manufacture
(d) the cost of keeping the product up-to-date
(e) the cost of maintaining the product after sale
(f) other costs not relevant to the choice of control method (marketing, shipping etc.)
In most instances, electronics will score over other control technologies in cost categories (a) to (e) above. But the line of argument can be pursued further to suggest what particular form of electronic control is likely to be most prevalent. The answer is **programmable microelectronics.**

The **design** cost of hardware is high, and the design and tooling costs of production equipment are also high. Mass-produced microprocessor controllers can be used in a number of different designs; the economics of scale keep down the hardware cost to a minimum. Different control programs ("software") will be needed for each application, and software costs are high too, but very often it will be cheaper to use a "standard" controller, (perhaps with more facilities than are needed) with customised software, than to design from scratch a new piece of hardware.

The raw material, or **component** cost, is very low for simple microprocessors - there is little to be gained from using cheaper components.

Manufacturing costs are unlikely to be much different once a plant is established. On the other hand, setting up an assembly line to make a new piece of hardware is very much more expensive than changing the software ingredient of one standard design that is used in a number of different control systems.

The cost of **modifications** to systems is dramatically lower for programmed systems - software changes are many times easier to make than wiring and circuit changes. This is especially true if systems already manufactured and sold to customers need to be modified or improved.

Microelectronic systems are very **reliable**, thus reducing maintenance costs. Again, the pressure is towards microprocessor systems because more of the electronics is "integrated" into chips, and even fairly simple controllers need a large number of components if a microprocessor is not used. The more circuitry that is integrated, the more reliable the system.

"Control Pathways" is about control by means of programmable microelectronic systems. As has been explained above, to a large extent any non-trivial control task is likely to be performed by a microprocessor or microcomputer. Pupils completing this course should be in no doubt of the versatility and flexibility of a programmable "control-heart". But an appreciation of advantages can only be gained when more than one method is assessed. What, then, are some other methods of control?

Switches and wires are about the simplest! There is no control in Section 1 that could not be performed by a human switching on the required lines, and leaving them on. This would probably take less time, and certainly be cheaper, than using a programmed controller. Sub-sections 4.1 and 4.2 use the controller as no more than a set of wires between inputs and outputs - again the system would not be used if this was all that was required of it.

Many control systems include mechanisms of necessity and sometimes complete control can be achieved by mechanical means. The first industrial revolution was essentially based on inovations in this field. The limitations of purely mechanical control are well illustrated by modern motor vehicle technology. Inevitably the total system includes mechanical sub-systems. But the **control** functions - engine control, anti-skid control, environmental control, etc. are increasingly performed by electronic rather than mechanical means.

Sequences can be produced **electro-mechanically. Relays** (large electro-mechanical current-controlled switches) could be used to perform several of the sequence functions in section 2. Delays can be implemented by wiring large capacitors across the relay. Many older traffic light controllers (upright boxes in the street at controlled junctions) work like this, and millions of relays are in use in telephone exchanges. The **cam timer** (a rotating shaft with cams on it which operate contacts in sequence) was until recently used extensively in automatic washing machines (for example).

For the more complex exercises in section 2 (where a counted number of pulses must be output), separate integrated circuits - counters, pulse generators, gates - could be used. In the hierarchy that is being considered, **hard-wired electronics** of this kind is next. It may be digital electronics (i.e. dealing with two states only, on and off) or ''analogue'' where voltages and currents can have any and varying values. For example, the ''Sunrise'' function (3.4) can be achieved digitally, or with analogue electronics.

It is also possible to use **memories** as control devices. The sequence of operations is stored in a memory; a pulse generator and counter ''addresses'' each location in the memory in turn. The stored control codes appear on the output lines which are connected to the controlled system. This is how the Movement and Music Modules are controlled by the Memory Module of MFA. Everything in section 2 could be done in this way.

Pneumatic systems can be used for control and can include inputs, delays and produce simple sequences. These had important safety advantages in fire-hazardous environments over systems using relays or electronic motors but the advent of the (non-sparking) solid state relay means that microelectronic systems can be used for the central controller in systems using pneumatic outputs.

Finally, **microprocessor systems** and **microcomputers** can be used - which is what this range of resources is all about. The control facilities offered are much the same but microcomputers are usually programmed in a ''high-level language'' which is easier to understand than the ''machine-code'' in which microprocessors must be programmed. Microcomputers are, however, more expensive, bulkier, and use more power.

The tendency is therefore to use the older technologies - mechanisms, pneumatics, hydraulics when required at the **output** of the control system. Characteristically these parts will be the most expensive and least reliable part of the total control system. The actual **processing of information** - ''if this happens do that'' - is increasingly performed by programmable microelectronic systems.

The engineer's business is that of **optimisation**. The teacher too will need to choose what is the best tool for a given control task. As the engineer will often (apparently) use a sledgehammer to crack a nut, so the teacher may guide the pupil towards the all-purpose programmable controller even for relatively trivial tasks. The advantage of having one control system that can do virtually anything is enormous and may outweigh the cost advantage of having some simpler controllers which can be used in selected applications. The speed with which programmed control can be implemented under classroom conditions is high, and there will certainly be times when a project outgrows a simple controller. In these circumstances anything other than the programmable controller will have to be jettisoned and design restarted from near the beginning. The use of a powerful controller from the outset will ensure that this does not happen.

As has been suggested, depending on the experience of the pupils, it is recommended that during their work on ''Control Pathways'' pupils be encouraged to consider alternative control methods which they have encountered in previous work for various of the problems.

6. Structured Flow Diagrams

Throughout Control Pathways, two types of diagram have been used to represent the algorithms employed to solve the control problems selected. The first is the standard flow chart which illustrates well the sequential nature of many control situations. When using modern well-structured languages however, its use is of limited value because there are no easy ways of representing the common loop and condition structures which form a central feature of such languages. One of the authors (JOL) has therefore devised a simple method of representing all such structures in a diagrammatic form which, it is hoped, will help those pupils using high-level languages to develop and understand their programs, and which may also be of use to those using machine code or assembler who wish to use a "top-down" approach to their program design.

Since structured flow diagrams are relatively new, it may be useful to explain some of the conventions employed. Firstly, every structured flow diagram represents either a program or part of a program (procedure) and always consists of a single structure box whose head (i.e. the upper part of the box) contains the name of the program or procedure. The body of the box (i.e. the rest) may contain either short statements about the processes to be carried out or other structure boxes. The statements are always read from top to bottom and in the simplest cases the structured flow diagram is identical to the equivalent flow chart (see the main pupil booklet).

Structure boxes are used to represent all kinds of loop structures, condition structures and procedures. In each case, the head contains a description of the kind of structure represented by the box. The following loop structures are used frequently:

REPEAT FOREVER

REPEAT n TIMES

REPEAT UNTIL ⟨condition⟩

In each case, the contents of the body of the box is carried out over and over again for the required number of times or until the condition is satisfied. E.g. Flash forever (sub-section 2.2), Alarm (sub-section 2.3) and Combination Safe (sub-section 4.3).

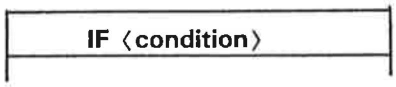
Two other loop structures are used but do not appear in Control Pathways. They are:

WHILE ⟨condition⟩

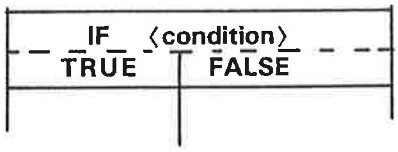
and

FOR n = a to b STEP c

Condition structures are represented in the same way. The simple condition in which a set of statements is carried out when a certain condition is true is represented thus:

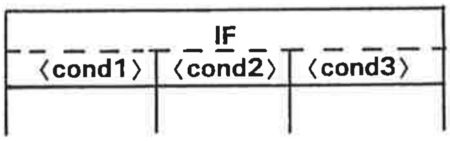


Many decisions are two-way, however. When the condition is true, one set of statements is carried out but, if the condition is false, a second set must be performed. An example of this is to be found in the program SWIM (sub-section 4.7). The body and part of the head is divided down the middle and the words TRUE and FALSE are added to the head to indicate which set of statements is to be carried out like this:



In order to emphasise the fact that only **one** set of statements is carried out, the two sets are written side by side.

The same idea may be extended to the multiple condition or CASE statement by dividing the body of the box into three or more compartments in the following way:-



In control work it is essential to consider the system as a whole - sensors, outputs, mechanisms, controller. The structured flow diagram is an aid to clarity of thought in designing the algorithm.

Structured flow diagrams can be used as a good way of disciplining one's approach to problem solving. The method can be exactly illustrated using the following specific example. Suppose we wish to program a simple "pelican" crossing. The "top-down" approach to this problem proceeds through stages.

Stage 1 - The total system

Exactly what must the system do? - Reproduce a standard system. What sensors are needed? - Something to serve as the usual push buttons; the 8 Switch Input. What outputs are needed? - the 8 LED Output will give us the traffic lights and the pedestrian control lights. If mechanisms or other structures are needed they must be designed in outline at this stage.

Stage 2 - The whole control sequence

We divide the sequence into obvious sections and construct an overall plan:

PELICAN

REPEAT FOREVER

Set traffic lights to green

Wait for several seconds

REPEAT UNTIL button pressed

Do nothing

Go through sequence

Stage 3 - Further analysis

Some of these sections are simple enough to be translated into the chosen language directly. The last statement is more involved, however, and needs further analysis. A second structured flow diagram or set of structured flow diagrams can be constructed and if necessary a third, carrying the analysis down to the point where it can be easily translated into the chosen language. It is important that, throughout these stages, the solution to the problem should be dictated **by the problem itself** and not by any consideration of the language or computer which is going to be used to implement the solution.

Stage 4 - Structural design

In this stage, we make decisions about the way we shall implement the structured flow diagrams in the language. Some languages do not always have all the facilities we should like and we may have to use the facilities which do exist to construct certain structures. For example early versions of BASIC had no REPEAT UNTIL facility and require the use of the less elegant IF ⟨condition⟩ THEN GOTO... It may also be necessary to set up certain counters or flags to keep track of what is going on and we need to decide on their names and initial values. When this is done we can proceed to:

Stage 5 - Coding and testing

Beginning with the simplest and most fundamental structured flow diagrams (which contain **no** structure boxes i.e. they do not call other procedures), we translate into the chosen language. When each structured flow diagram has been translated, it should be thoroughly **tested** to make sure that it works as intended. For example, the sequence which changes the lights would be written and tested first in our example above.

Stage 6 - Writing the "main" program

Finally, when all the components have been written and tested, writing the actual program is trivially simple and should work first time! (Though it is essential to test it when complete.)

The cycle is complete. We start with the whole problem; break it down into smaller and smaller section, then build it up as a coded program, testing as we go until the program is complete and the problem is solved. As a medium through which to express and refine your ideas about the solution to a control problem, the structured flow diagram is recommended, particularly when high level languages are used.

7. Worksheets

Notes for the Teacher

The worksheets in this section are intended for photocopying and should be made available for pupils to stick into their exercise books as indicated in the main pupil booklet.

The purpose of the worksheets is three-fold. The first is to reinforce the concepts developed by pupils in their work. The second is to encourage a structured form of recording of some of the core material as an aid to reference. The third is to aid the teacher in assessing the progress of individual pupils so that any misconceptions can be clarified. Note that, because section 1 is very short, there is no worksheet number 1.

WORKSHEET NUMBER 2 - SEQUENCES

W2.1

Write down two examples of electroncally controlled sequences in your home or school:-

(i) ______________________________

(ii) ______________________________

Write down the command(s) you used to output 1111 1111 (binary)

W2.2

Write down the command that makes the sequence repeat continuously:-

____

____

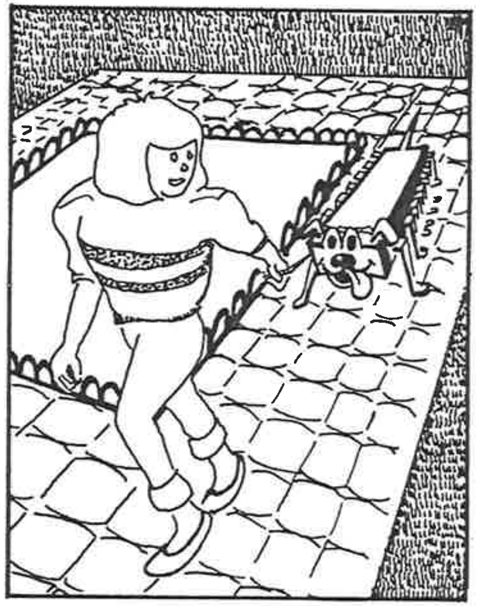
W2.3

The original program sounded a note 20 times. You were asked to change one command to make the note sound 5 times.

Write down the command you used to make the note sound 5 times:-

____

W2.4



How far did your buggy travel forwards in 4s?

____

What delay did you use to make your buggy travel 0.4m?

____

What command(s) did you use to produce this delay?

____

____

W2.5

Complete this table:-

Lights on Major road	Lights on Minor road	Binary Output
Green	Red	00 001 100
Amber	Red & Amber	00 010 110
Red	Green	.. ..
Red & Amber	Amber	.. ..

What command(s) did you use to give Amber on the Major road and Red and Amber on the Minor Road?

____

____

WORKSHEET NUMBER 3 - D TO A CONVERSION

W3.1

What must the controller for the MAGLEV system do to the vehicle's speed as it moves between stations?

Why is this different from the Traffic Lights or the "Alarm" system?

Circle the correct word:-

As the voltage increases the brightness of the lamp **increases/decreases.**

W3.2

Circle the correct words:-

Changing over the negative and positive connections causes the motor to **change direction/go much faster**

W3.3

Fill in the table headings:-

Your height	Output of AND gate
Strength of wind	Computer memory contents
Speed of a rocket	LED pattern

Write down two more types of DIGITAL INFORMATION

Write down two more types of ANALOGUE INFORMATION

W3.4

Complete this table:-

Digital output from controller to D to A Converter	Voltage output of D to A Converter
0000 0000	0
0000 0001	
0000 0010	
0000 0011	
0000 0100	
0000 0101	
0000 0110	
0000 0111	

Bits 4, 5, 6, and 7 are not connected to the D to A Converter and do not affect it.

W3.5

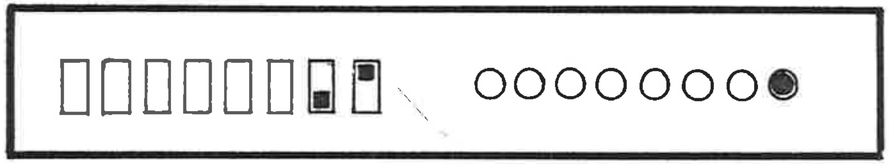
Complete this table:-

Digital output from controller to D to A Converter	Voltage output of D to A Converter
0000 1000	
0000 1001	
0000 1010	
0000 1011	
0000 1100	
0000 1101	
0000 1110	
0000 1111	

WORKSHEET NUMBER 4 - INPUTS

W4.1

Complete the picture below for an input of 1010 0001.



This switch pattern gives this LED pattern when the controller copies the INPUT to the OUTPUT.

W4.2

What did you change the combination of the safe to (in binary)?

____

Write down the command for your controller which set this new combination:-

____

W4.3

Write down, for your controller, the command which would give a combination of 1001 1001.

____

W4.4

Make a note of the line in the program (from section 4A of your mini-manual) which reads the **input** to your controller:-

____

W4.5

Write down two situations where it might be useful to count people going through an entrance and in each case, explain **why** it might be useful.

First situation:-

____

It would be useful to count people because:-

____

Second situation:-

____

It would be useful to count people because:-

____

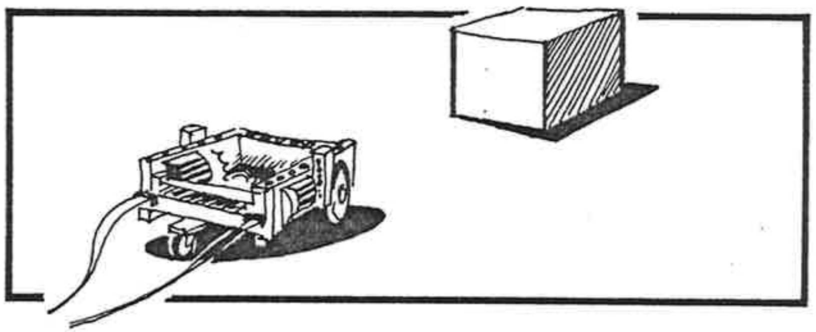
WORKSHEET NUMBER 5 - FEEDBACK

W5.1

Write out the commands you **added** to your program to make the buggy go backwards for 1s and then stop.

W5.2

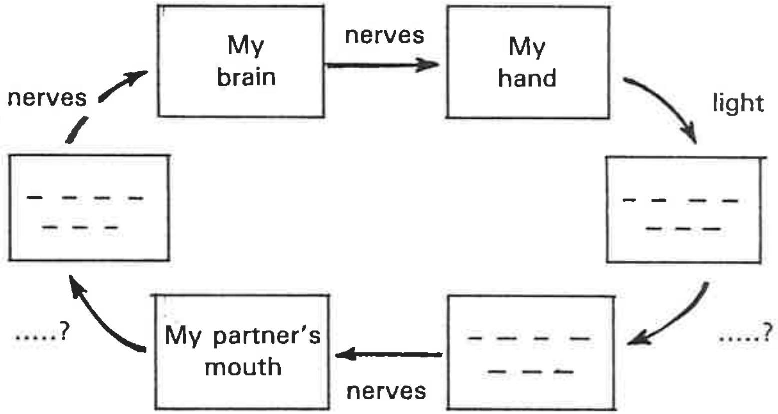
Sketch the path taken by the buggy when it hits an obstacle. Label the corners with the delays you used so that the buggy turned through 90°.



W5.3

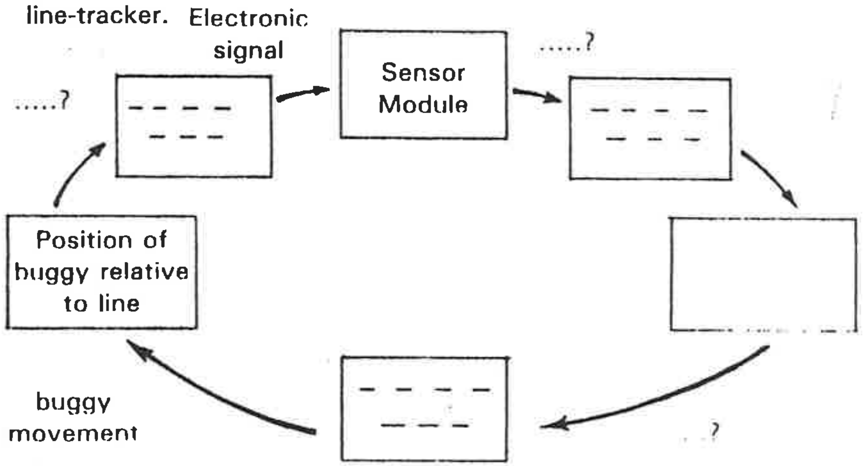
Look at the diagram in the main booklet showing the "feedback loop" when you joined the points with your eyes **open**.

Now complete this diagram which shows the feedback loop when you worked with your partner with your eyes **closed**.



W5.4

Complete a similar diagram for the "feedback loop" of your line-tracker.



WORKSHEET NUMBER 6 - A TO D CONVERSION

W6.1

Adjust the potentiometer until the voltage input to the A to D converter = 0.36V (the first value in the table). Enter the binary value shown on the LEDs in the second column of the table. Work out the decimal ("counting in tens") value of the binary number shown by the LEDs. Write this into the third column of the table.

Then go on to the second line of the table - adjust the potentiometer, read the LEDs etc.

Voltage input to A to D Converter	Binary value on LEDs	Decimal value on LEDs
0.36V		
0.83V		
1.46V		
1.98V		
2.37V		
	1111 1111	255
	0000 0000	0

The last two lines in the table are different - adjust the potentiometer to get the LEDs all on, and then all off. Read the voltage input for each - these are the maximum and minimum input voltages that the system can handle.

WORKSHEET NUMBER 6 - A TO D CONVERSION

W6.1

Adjust the potentiometer until the voltage input to the A to D converter = 0.36V (the first value in the table). Enter the binary value shown on the LEDs in the second column of the table. Work out the decimal ("counting in tens") value of the binary number shown by the LEDs. Write this into the third column of the table.

Then go on to the second line of the table - adjust the potentiometer, read the LEDs etc.

Voltage input to A to D Converter	Binary value on LEDs	Decimal value on LEDs
0.36V		
0.83V		
1.46V		
1.98V		
2.37V		
	1111 1111	255
	0000 0000	0

The last two lines in the table are different - adjust the potentiometer to get the LEDs all on, and then all off. Read the voltage input for each - these are the maximum and minimum input voltages that the system can handle.

8. Project Possibilities

Notes for the teacher

The main pupil booklet contains short notes on project possibilities. The extended notes in this section are intended to be given to pupils when they have selected a project, or a short list of projects.

Some pupils will have their own project ideas; if so they should be encouraged to carefully write down what they plan their system to do and check what inputs and outputs will be needed.

It is strongly recommended that time be found for at least most pupils to tackle a project. From limited experience I would guess that this will be the high-point of the work and create a lot of interest and discussion.

You will need to counsel the pupils in choice of project (they vary considerably in difficulty) but the interest and enthusiasm of pupils is probably the best guide.

You will need to keep account, as each project is agreed, of the peripherals and kit systems "used up". While pupils will not need these all the time (the actual program can often be tested using just the input module and the 8 LED output) it would be difficult if all pupil groups tried to use a peripheral in short supply.

As pupils develop their ideas you should encourage them to write each part as a **subroutine** or **procedure** (see section 6 of this manual). If they try to write the entire control program as a single program they will make many mistakes, find serious difficulties in checking for errors and not be able to build up and improve their ideas.

The "Output T-piece"

This part is not used in sections 1 to 7 of the main pupil booklet but is needed for several projects. It allows the three output modules which cannot be followed by other standard output modules (the Movement Module, the Movement Module and the Relay Unit) to be used together in any paired combination e.g. Relay and Music, two Movement. In addition the Output T-piece allows the bit-assignment of outputs to be changed. In consequence any of the output systems using four bits can be used in any paired combination e.g. D to A Converter plus Stepper Motor, two Music Modules.

The Output T-piece connects bits 0 - 3 of the controller output to bits 0 - 3 of its lower output connector **and** to bits 4 - 7 of its lower output connector. Thus bit 0 of the controller output is connected to bit 0 **and** to bit 4 of the lower output connector etc. It also connects bits 4 - 7 of the controller output to bits 0 - 3 **and** to bits 4 - 7 of its upper output connector. Thus bit 4 of the controller is connected to bit 0 and to bit 4 of the upper output connector. So, **provided** an output which normally uses bits 0 - 3 of the output is connected to the lower output connector and an output which normally uses bits 4 - 7 is connected to the upper output connector, the bit outputs needed will be the normal ones.

If it is necessary to use together two outputs which normally use the **same** four output bits e.g. the Movement Module and the Music Module, this can be done **but** the output attached to the upper connector will be controlled by bits 4 - 7 of the controller and the output attached to the lower connector will be controlled by bits 0 - 3.

"Simon"

"Simon" is an electronic game. The manufactured version is circular (0.3m diameter); the four quadrants of the circle are integral switches and lamp-lenses. The machine lights the four (different coloured) lamps in a random sequence. The player must press the switches in the same sequence as Simon lit the lights. So, Simon (perhaps) flashes red, you press red. Simon flashes red, yellow; you press red, yellow. Simon flashes red, yellow, red ... so do you; red, yellow, red, green ... you follow. The aim is to copy as long a random sequence as possible. Simon also makes noises at the same time as lighting a light - a different tone for each light, plus a raspberry when you get the sequence wrong!

Connect the 8 LED Output and the Music Module to the output of your controller. It is possible to control LEDs 4-7 without affecting the Music Module output. So bits 4-7 can be used to control Simon's lamps (the LEDs on the 8 LED Output), and bits 0-3 to control tone, without interaction. It is probably helpful to cover LEDs 0-3 so that they can't be seen. Put the 8-Switch Input on the input. Switches 4-7 must be switched in the same order as LEDs 4-7 flash. It would be straightforward enough to make a real "Simon" unit with push switches alongside lamps.

Generating a random number between 0 and 3 is easy with some controllers, not so easy with others - especially machine-code controllers. The key presses of the player are events that happen at random times; these could be used to generate random numbers. After Simon has output his sequence, a machine-code controller will go into a loop waiting for an input switch to be pressed. If the contents of a memory location are incremented each time it goes round the scanning loop, a random number will be left in that memory cell dependent on the instant the key is pressed. If the content of the cell is ANDed with 03, the result will be a number between 0 and 3 - the next "random" number for the sequence. That number can be used to select LED 4, 5, 6, or 7.

For those who enjoy programming to meet a desired end, this is potentially a rich project.

Sophisticated Doorbell Systems

The simplest idea is a doorbell that will tell whether the visitor is at the back or front door. A different tune could be played when bits 0 and 1 on the input are switched. Obviously 8 different bellpushes could be accomodated!

A system could be devised that changes the tune each day (what ways are there of detecting a day-change?).

What about a doorbell that plays a tune which speeds up, or slows down(!), when the temperature rises?

It might be desirable to switch the bell off at night to discourage late-night visitors. But in winter it gets dark at 4 p.m. In the absence of a real-time clock to which to connect the controller, it might be possible for the system to discover what time of year it is, and delay "switch-off after dark" accordingly.

How could the system know when the house is empty? It could refuse to ring the bell under these conditions, or sound a "the resident is out" tone. But then, would it be desirable to monitor whether the caller has gone away within 30 seconds, and sound an alarm if not?

Projects that can start simple but have a lot of growth potential have much to commend them; this is one such.

Washing Machine

A typical washing machine has the following sensors:-

1) DOOR - senses whether the door is open or shut
2) WATERLEVEL - senses whether the water is at the correct level or not
3) TEMPERATURE - senses whether water is at the correct temperature or not

It also has the following outputs:

1) WATERIN - this controls a valve which lets the water in
2) HEATER - this switches the electric heater on and off
3) MOTOR - this controls the motor
4) WATEROUT - this controls a valve which lets the water out

Your controller has enough inputs and outputs to do all of these things. Connect the Sensor Module to the input side and connect a microswitch (for DOOR), a level sensor (for WATERLEVEL) and a temperature sensor (for TEMPERATURE) to it.

On the output side connect the Relay Unit.

You may have more difficulty in devising the plumbing but, if you can get them, connect two valves to two of the relays, a small heater to a third relay and a motor to the fourth relay.

Note that these will all need to work from the **same** voltage so check this.

A simple washing sequence would be:-

Fill drum
Heat water
Agitate
Empty drum
Fill drum
Agitate
Empty drum
Spin

Translate **each** one of these into a short flow chart or structured flow diagram and then write a program to accomplish this task. Then write a main program to call them in sequence. Modify your program if you can so that, if at any stage the door is opened, the cycle stops.

Security System

An intruder-alarm is to be provided for a museum. Two operative modes will be available - daytime when the museum is occupied, and night time when it is shut.

Sensors will cover exhibition cases, doors, floor space and alarms will be situated in the security office (day) and a police station and outside the wall of the building (night).

If possible, indicator lights will show which sensors have been triggered. A delay will be inbuilt between night time activation of police and exterior alarms.

Model Railway Control & Wagon Coding

A layout with 4 points and 4 sensors can be controlled. It doesn't matter whether it is a small "N" gauge layout or a larger "OO" gauge layout. Loops can be used, or straight track with sidings at each end.

A train (or just a locomotive) can be taken on a route with speed and point control codes, delays and sensor "masks" (which tell which sensor is to be scanned next) in the main program or, better, stored as a data table. A timetable can be operated and changed relatively easily (especially if a data table is used for the control information).

An alternative is to write a program that scans all sensors continually and routes the train according to where it is detected - this can be useful if unreliable points are used in an exhibition. If the train appears in an "unexpected" position the system can cope with it.

If your controller is a computer with a VDU you can have a mimic diagram on the screen which tells the state of all the points and where the train is.

A marshalling yard can be built with a number of sidings running down a gentle slope. Wagons (it's easier with the larger OO wagons) may be coded with small pieces of cooking foil stuck on the side, and read with a reflective opto-switch. One way is to have two lines of squares, and two opto-switches. The "marker" line is simply

SQUARE/space/SQUARE/space/SQUARE/space

(for a three-bit code). This tells the system where squares should be. The other is the code line, which might be

SQUARE/space/SPACE/space/SQUARE/space

This would code 101. The program regularly interrogates the "marker" opto-switch to check that a truck has not appeared yet. When a truck is detected, the code line is read when a SQUARE is present in the marker line.

The coding and reading system described has the advantage that it is fairly reliable whatever speed the wagon goes past. But it does use two sensors, and the wagon can go past in one direction only. There is a great deal of work that could be done improving the system (and a good deal that can be learnt from work on barcode reading).

No electric uncoupler exists for N gauge (to this author's knowledge). A spare bit will have to be used in any case to switch the uncoupler so you might only be able to use 3 points. Manual uncoupling may be the best answer, if the least elegant.

Robot Arm

A robot arm is programmed to detect the presence of steel billets on an input work store. If they are there, it electromagnetically lifts them, then places them under a (pneumatic?) stamping machine. On completion, it removes the billet and places it in a finished work store, and repeats this until there is no more input work.

A sensor halts the cycle if a person enters the working zone, and triggers a safety alarm.

Two motors will operate the arm in the horizontal and vertical planes.
(The output T-piece would be useful)

A Paper Tape Organ

Fairground steam organs have their tunes stored on wide cards with holes punched in them. Could a model of an improved system be built?

A four-hole code (across the paper) could be used. This binary pattern could control the note or its length. Perhaps alternate codes carry note and length information. The control system then reads in a code, sounds the note it indicates, reads in the next code, and waits for that number of delay units.

A mechanical system will have to be built that will advance the tape across a row of four opto-sensors. Slotted opto-switches will be hard to use for the middle holes; separate LEDs and photodiodes or reflective opto-switches will be easier (which can be connected to the Sensor Module). The controller could switch on the drive motor until the next line of holes appeared, stop, read, move on to fetch the "length of note" code, delay while the note is played, and so on.

If your tune sounds the same note twice (or more) in succession you need to have a short "silence" in between the notes.

You will need an "Output T-piece" to enable you to connect both the Music Module and the Relay Unit (or the Movement Module if you are using a 5V motor) to the controller output at the same time. Bits 0 - 3 can then be used to control the sound and any of the bits 4 - 7 to control the tape drive.

Can the Music Module sound 2 notes at once? What happens if middle C and A (frequencies of 261 Hz and 330 Hz) are switched 20 times a second? It may be that the ear will hear something like a chord if 15 cycles of one note are sounded, then 15 cycles of the other, then 15 of the first, etc. Write a short trial program to control the Music Module directly in this way (put the tune in the program, don't worry about paper tape input). If it sounds reasonable, modify the organ to play harmonies.

What other coding systems are there? What are the advantages and disadvantages (in theory and in practice) of a one-hole "serial" code? (Four holes across the tape is a "parallel" code; four holes one after the other is a "serial" code.)

You will have to search carefully for tunes because the range of the Music Module is limited.

Central Heating Controller

Microprocessors are being used for environmental control and this system provides simple control of two heat sources - a gravel heat store (heated up from solar panels) and a gas boiler. Room sensors and an external sensor provide inputs. Solenoid operated dampers and air ducts control the flow of hot air to the rooms. An inbuilt program sequences heating to various parts of the house during the day to provide the optimum temperatures in various rooms according to occupancy.

Heat will be used from the gravel heat store unless it is too cold, then the gas boiler will be ignited.

Puppet Show

A good puppet theatre needs a lot of skilled human control. But if you want to have a simple display to grab attention (in the school entrace or a local shop to advertise your next theatrical masterpiece?) you could use your controller to give a sequence again, and again, and again... for days.

Your controller has eight output bits. These can control four motors (forward and reverse). These can be 5V motors, controlled by the Movement Module or 12V motors controlled by the Relay Unit. If you only need two motors (controlling two strings) you can use either unit. If you need four motors you will need the "Output T-piece" which divides the 8 bits of output into four and four.

Each motor can go clockwise, anti-clockwise or stop. Study the details of the Movement Module and the Relay Unit. **Different** output patterns are used for these modules to give these results.

The strings will move the puppet. So think about any gears or pulleys you may need to control the speed of movement of the strings.

It's probably best to have a "rehearsal" with a human puppet controller, **before** you develop the program, to sort out a really effective sequence. Measure distances and times. Remember you can make the repeated sequence quite long and complicated.

What about some inputs - different dances for day and night, or start the dance when someone pushes a button or change the speed of the dance depending on the light level?

Or what about other outputs - two strings (from the Relay Unit) and a tune (from the Music Module)? Or three strings and a spotlight (controlled from the Relay Unit)?

Four strings is a bit limiting. If you're using 3-Chip Plus you could use both ports as outputs and have 8 strings, or 6 strings and two spotlights or.......

Conveyer Belt Sorter

In many factories, components are checked as they pass along a conveyer belt and if they are faulty they are removed. You could build a conveyer belt to carry small plastic or wooden bricks. By positioning a sensor at a certain height above the conveyer and an arm at the side operated by a motor attached to the Relay Unit, you could program it to knock off all the tall blocks and leave the short ones.

How would you modify this arrangement so that the controller knocks off the **short** ones and leaves the tall ones?

Suppose the bricks were of different length, not height. How would you get your controller to sort the long bricks from the short ones?

Can you sort different coloured blocks? Could you have two arms to knock off blocks and sort them into three groups?

Pneumatic Control

You can use these ideas if you already have experience of using pneumatics for control.

You will need several solenoid-operated valves.

It is possible to design pneumatic systems to carry out automated sequences, for example:- extend Cylinder A, delay, extend Cylinder B, delay, withdraw Cylinder B, delay, withdraw Cylinder A, delay and repeat the cycle. (Further examples are given in books on pneumatics e.g. Introducing Pneumatics by Chris Weaving, produced and distributed by Teaching Media Resource Service, Russel House, 14 Dunstable Street, Ampthill, Bedford, MK45 2JT).

Instead of using flow restrictors and reservoirs for the delays you can use your controller. The 3-port or 5-port valves have to be operated by a solenoid (you can use the Relay Unit to power them - **carefully check the voltage needed**). You can use microswitches and the Sensor Unit to start the sequence and to detect the position of the Cylinders.

Set up the **same** sequence

(i) Using pneumatics alone

(ii) Using your controller and pneumatics

Now write a report and assess the advantages and disadvantages of the two methods. Consider - cost, time needed to design systems, ease of modification, safety, reliability and anything else you noticed.

Chart recorder/X-Y plotter

Scientific experiments often require a graph to be plotted of an analogue value against time (a chart recorder) or of one analogue value against another (an X-Y plotter).

The analogue value could be temperature or light - the signal from the sensor being fed to the controller through the A to D Converter. Or you could feed in a voltage from **any** source on the 4mm inputs. If you have access to a "System Alpha" electronics kit you can input the signal from this by attaching it to the side of the A to D Converter.

Stepper motors could be used to move the pen over the paper both along the X axis and Y axis. Alternatively one motor could be used to move the pen along the Y axis and a second motor could be used to move the paper along the X axis.

You can use two Stepper Motor Driver boards connected together - one controlled by bits 0 and 1, the other by bits 2 and 3.

One motor should be stepped at regular intervals to give "time" the other motor should be controlled according to the size of the analogue value (for a chart recorder). For an X-Y recorder both motors are controlled according to the two analogue input values and you will need the analogue multiplexer. The signal for this will need to come from bits 4 - 7 of the output so you will need to arrange special wiring for this.

Communications

Here are some suggestions if you have an idea for a project where the system input needs to be distant from the system output. You can arrange for the system input to go into one controller, and for this controller to switch off and on a torch bulb (with the Relay Unit). Then you can send the light to a light sensor attached to a Sensor Module on a **second** controller which can then feed the system output. Be very careful to shield your light sensor from stray light.

How can light be used to send information? One way is called **pulse width modulation.**

The torch bulb is switched on for a length of time which varies, depending on the information. The 8 binary bits of the input can represent numbers from 0 to 255 (decimal). You could arrange for the lamp to be on for this number of "delay units". (What will you do for 0?) The program on the receiving controller measures this time. While the lamp is off the first controller reads the input and the second controller sets the output. Be careful about the time your controller takes to carry out **other** commands (not just the delay).

A more efficient method is **pulse code modulation.** You can read about this and other communication systems in "Telecommunications in Practice" edited by B Nicolls and J Self and published by British Telecom and the ASE.

Automated Fish Tank

A system is needed to control the environment of a tropical fish tank.

a) Temperature
b) Air
c) Light
d) Feeding

These four variables need to be controlled over a twenty-four hour period and this sequence needs to be repeated accurately for several weeks.

The temperature needs to be kept constant. Feedback from a temperature sensor could be used to control a heater.

The air pump could be switched on for five minutes in every fifteen providing adequately oxygenated water and saving energy.

The light could be switched on for eighteen hours in every 24 to simulate tropical conditions.

A feeding device needs to be developed which could be activated at regular intervals. Think carefully about the design of the mechanism for the feeder.

Perhaps you only need put the light on when it's dark.

Drilling Rig

A system is needed to control the position of the workpiece.

Stepper motors offer positive drive for carrying out this task.

Stepper motor 1 could move the table from left to right, stepper motor 2 moves the table forwards and backwards. You can use two Stepper Motor Driver boards, connected together - one controlled by bits 0 and 1, the other by bits 2 and 3.

Accurate positioning is carried out by causing the motors to turn through a number of steps.

The depth of the hole could be controlled by a micro-switch whose position is adjustable. A third motor, an ordinary 12V DC motor, can be used to move the drill up and down. You will need to use the Relay Unit to switch off and on this motor and the motor of the drill itself. Make sure these take the same voltage (or else figure out how to allow for the different voltages).

Programs could be developed to drill several holes in the same workpiece.

If you are using a computer, could you arrange for the user to type in the hole positions?

Coin Dispenser

A coin dispensing machine has a stock of 1p, 5p and 20p coins. It is programmed to give change for any amount from 1p to 99p. The amount required (from 1p to 99p) could be set up on bits 0 to 6 on the 8 Switch Input. Bit 7 could be used to indicate when the correct amount has been set up on the switches and when delivery is required. On the output side, you will need three levers to deliver the necessary coin.

One problem you will have to solve is how to work out the minimum number of coins required. It can always be done with 11 or fewer coins.

Could you arrange that the system only gives coins if the correct combination is input first (like the combination safe)?

Or could your system detect the size of an **input coin** and give the right number of the next smallest coins?

Complex Traffic Lights

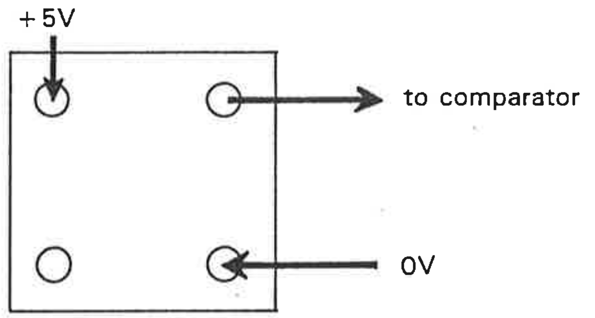
There is a busy junction in Computer Town where Digital Drive crosses Analogue Avenue. Digital Drive has a pedestrian crossing where Milli-Amp and Micropup cross to get to Silicon Secondary School. There is a button for Milli to press and there are sensors in both roads to sense when traffic is waiting. Digital Drive usually has more traffic on it than Analogue Avenue so if there is no traffic on Analogue Avenue after 10s, the traffic lights change back to Digital Drive. Traffic on neither road should have to wait longer than 60s though for the lights to change. Nor should Milli-Amp. Of course, when the pedestrian lights go green, all the lights should remain at red for about 10s.

Do you think you could develop a system to control these traffic lights? It is not really as difficult as it looks, **provided** you spend a great deal of care getting your flow chart or structured flow diagram absolutely right. Use the 8 Switch Input and the 8 LED Output.

A Bolt-on Goody

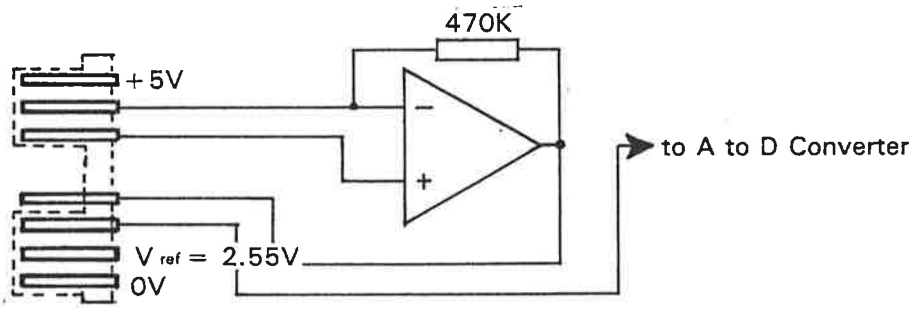
This isn't a complete project but could help if you've got a project idea that needs a sensor or an output device that isn't available in the standard kit. To do this you will need to know a bit about electronic circuits.

The easiest task is a new input sensor for the Sensor Module. The connections on the Sensor Module are:-



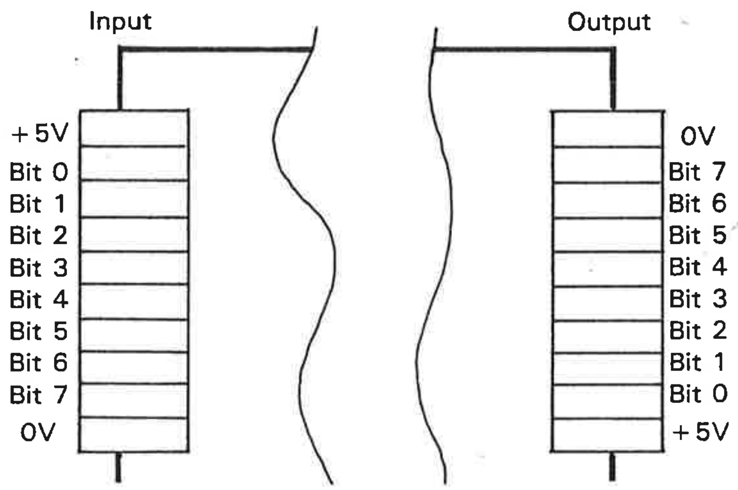
So you can arrange a potential divider with a resistor and your chosen sensor, with the mid-point going to the comparator. Suitable sensors, not available in the standard kit, are a mercury tip-over switch, a reed switch and a float reed switch. You can also use a Hall effect IC switch, a linear or rotary potentiometer and a light-activated switch with this input. So you could have projects with inputs from a changing tilt, a magnet, a non-aqueous liquid, rotation or linear movement or force.

The input to the A to D Converter is a bit more complex. The connections are:-



A link has to be made from pin 3 to pin 4 to connect the operational amplifier output to the A to D Converter. One interesting sensor that is not in the basic kit is a microphone.

If you want to be really adventurous you could think of complete new input or output modules. The connections from the controller are:-



The output will drive one standard TTL load and the inputs present one standard TTL load. All connectors are available from UNILAB.

Alarm for Disabled or Aged

When an old or disabled person lives on their own, it is possible for them to become ill, or have an accident, and not be able to call for help. The aim of this project is to design and build a suitable control system to raise the alarm if the normal pattern of living is not being followed.

You must decide what events constitute a ''normal'' living pattern. How many of these events can be checked electronically?

Possibilities are:

Use of cooker
Opening of doors
Curtains opened during the day
Toilet flushed
Bed slept in
Temperature suitable during the day

You must then think of suitable sensors which will warn when the normal pattern of living is **not** being followed. Some possibilities are:

Movement detectors - microswitches which detect the opening of a door, or pressure mats on the floor.

Room temperature sensing - it must not become too cold, especially during the day.

Time will be important in your alarm system. How long should the alarm system wait if movement is not detected in a morning?

What if there is movement, but it is still very cold?

If the cooker has not been used for 10 hours should the alarm be raised, or would it be better to wait for 24 hours?

When you test your program you will need to shorten the timescales. Perhaps 10 seconds to represent 1 hour will be suitable.

What will you use to signal an alarm? In practice, the output of this type of system could be used to switch on a British Telecom ''CARE'' 'phone. This 'phones four memorized numbers in turn until a reply is received. It then delivers a recorded message.

An automatic Greenhouse

For optimum plant growth, the temperature, humidity and soil moisture in a greenhouse must be maintained between narrow limits. Also plants must be fed, usually using liquid nutrients.

You should design a control system to do this, and anything else which you think would be useful.

For research work on plant growth it might be useful to measure the total light falling on the plants. Or use a movement sensor to detect and record plant growth, together with a light sensor, to see if the rate of growth depends on light. How does moisture, or feeding, affect growth?

Possible devices for your system to control, are:

A heater
An electric motor to open windows
A solenoid operated valve to switch on water and/or nutrient

Possible sensors are:

A humidity sensor
A soil moisture sensor
A temperature sensor
A light sensor
A movement sensor
Sensors to indicate if windows are open or closed

Solar Panel Tracker

The aim of this project is to build a model with an aerial which can be moved by electric motors in two directions at right angles. It should then be able to "follow" a light source as it is moved slowly around it.

It will be easier to build a system where one motor rotates the aerial about a vertical axis and the other tilts the aerial from one side to the other. However the software to drive this system will be more difficult.

Light sensors could be arranged in two positions round the aerial. If more light was falling on one that the other, then the motor would be switched on to drive the aerial towards the sensor with more light.

CAN YOU THINK OF A BETTER WAY?
This will not be an easy project!

A Learning Robot

Many industrial robots are ''taught'' by human beings how to do a particular job. The robot then continues to do the same job, in exactly the same way, for as long as necessary.

An example of this is a robot paint sprayer. When a new object is to be sprayed, the robot arm is set to a ''learning'' phase. A skilled painter then moves the arm, with paint spray attached, to paint the object. The robot remembers every move, and so can repeat the operation as many times as needed.

You should design and build a robot which can be ''taught'' from the keyboard. Unfortunately this project cannot be done with 3-Chip Plus because it does not have a separate keyboard.

The robot must have position sensors (potentiometers) to detect the rotation of the arm, and also the up and down movement of the arm. These can be input to the A to D Converter, through the multiplexer. Double-sided sticky pads are useful for sticking the base of a potentiometer to your model.

The motors must be able to reverse. These can be driven from the Movement Module, or the Relay Unit.

First decide which keys to use to steer the motors. The cursor keys can be used, or ''U'' for up, ''D'' for down, etc. Then program the computer so that pressing ''U'' starts the ''up'' motor, and pressing it again stops it.

You then need to store the key presses in an array, and also store the reading of the position sensor when the motor stops.

After storing a sequence of movements, you then need a ''replay'' program. This reads the stored key, and starts the appropriate motor. It then inputs the changing reading from the position sensor, until it reads the same as the stored value. It then switches on the next motor, and repeats.

A good task is to make a robot transfer an egg from egg-cup to pan. You may need 3 motors for this in which case you would need to use the ''Output T-piece''

9. Assessment

Notes to the teacher

Four schemes of assessment are given as possible models. Teachers are encouraged to adapt these to suit their own approaches.

There is one point that must be made emphatically.

THE CENTRAL AIMS OF CONTROL PATHWAYS WILL BE FRUSTRATED IF AN ASSESSMENT SCHEME IS USED WHICH REQUIRES DETAILED RECALL OF PARTICULAR FEATURES OF PARTICULAR CONTROLLERS e.g. RECALL OF DETAILS OF LANGUAGE COMMANDS.

Both the written tests are intended as **OPEN-BOOK EXAMS** and pupils should have access during the exams to their exercise books, the mini-manual for their controller and the main pupil booklet and they should know before the exam that they will have such access.

The purpose of the assessments is to establish to what extent pupils can tackle realistic control problems. That capability has very little to do with **recall**; it is primarily to do with **systematic logical problem-solving**. No control engineer in industry tackles control problems by first locking away the information manual on the controller. The assessment scheme adopted should reflect this reality.

Assessment Model A is a scheme of **self-assessment** for pupils to monitor their own learning. It is recommended that this be used in all cases. The pupil completes the left-hand column with a tick when he/she has successfully completed the section. The teacher then either endorses this (in the right-hand column) or asks the pupil to demonstrate competence in the skill involved.

Assessment Model B is intended to allow assessment of the Educational Aims (see section 2 of this manual) numbered 1, 2 and 3. These aims are relatively "ephemeral" and the check list is provided as a possible method of gauging the fulfilment of these aims for pupils.

Assessment Model C is intended as an end-of-course partly practical, partly paper-based assessment. There are logistic problems in such an exercise, depending on the number of pupils and amount of apparatus involved. Nevertheless a partly practical-based assessment is preferable if it can be achieved. For the practical work the controller should be set up with a **selection** of input and output devices available.

Assessment Model D is intended as an end-of-course paper-based assessment. In question 5 it will be necessary to blank off either the flow chart or the structured flow diagram (depending on which system the pupils used) and to write alongside the system actually used a program for the controller used by the pupils.

SELF-ASSESSMENT EXERCISE ASSESSMENT MODEL A

	PUPIL	AFTER DISCUSSION WITH TEACHER
Section 1		
1. I know how to output a given 8 bit pattern from my controller	☐	☐
2. I can change one bit of the output at a time.	☐	☐
3. I can control a two motor buggy from my controller.	☐	☐
Section 2		
1. I can set up a given delay in a program.	☐	☐
2. I can send a given sequence of binary patterns to an 8 LED Output.	☐	☐
3. I can make the Music Module buzz 20 times.	☐	☐
4. I can change the number of buzzes of the Music Module, and understand how to control the number of times a program goes round a loop.	☐	☐
Section 3		
1. I understand what is meant by an analogue quantity. I can name several analogue quantities.	☐	☐
2. I understand what is meant by a digital quantity. I can name several digital quantities.	☐	☐
3. I understand why a D to A Converter is needed if a microprocessor is to control the speed of a motor, or the brightness of a lamp.	☐	☐
4. I can use the D to A Converter to make a light bulb gradually come on or gradually go off.	☐	☐

Section 4

1. I know how to program my controller to read an input from the 8 Switch Input. ☐ ☐

2. I understand how the input can be used to switch on output lights or motors. ☐ ☐

3. I understand how the controller can be used as a simple "combination lock", and know how to change the combination. ☐ ☐

4. I understand how sensors can be used as inputs to the controller. ☐ ☐

Section 5

1. I can make the buggy stop when it hits an obstacle, or dodge an obstacle. ☐ ☐

2. I understand why pressing the microswitch bumper can stop or change the direction of the buggy. ☐ ☐

3. I can explain why feedback is needed if dots on a piece of paper are to be successfully joined, or a key is to be placed in a lock. ☐ ☐

4. I can make a buggy follow a line. ☐ ☐

5. I can understand why it would be impossible for the buggy to follow the line without feedback. ☐ ☐

6. I know how information about the position of the buggy relative to the line is fed back to the controller. ☐ ☐

Section 6

1. I know what an A to D Converter does. ☐ ☐

2. I can use the A to D Converter and a sensor to input an 8 bit number into the controller. ☐ ☐

3. I can use the controller as a digital thermometer. ☐ ☐

4. I know that the resolution of the A to D converter I used was 0.01 Volts. I understand what is meant by the resolution of an A to D converter. ☐ ☐

ASSESSMENT MODEL B

This scheme is an attempt to assess subjectively the listed aims 1 to 3 of the Control Pathways materials.

It is suggested each pupil is assessed in terms of a series of statements on a 1 to 5 scale for each of the statements below. Meanings of the numbers are:

1 strongly disagree
2 disagree
3 not sure
4 agree
5 strongly agree

It would probably be worth doing this assessment several times during the course.

Statements:

A The pupil was interested in the work.

B The pupil was largely successful in solving the problems

C The pupil discussed control problems with his/her partner, and contributed ideas to help solve the problems.

D The pupil co-operated well with other members of the group in solving problems.

One way to record answers is to produce a form with pupils' names down the left-hand side. The letters for the statements could then be placed across the top, and numbers written in for each statement for each pupil.

A similar method might also be useful in assessing written records. Suitable statements may be:

E The written record is clear and complete.

F Good understanding is demonstrated.

G Time has been well spent.

ASSESSMENT MODEL C

1. Describe, in a logical order, how you would make a cup of tea.

2. Draw a flow chart or structured flow diagram for buying a bag of sweets.

3. You are to design a system using your controller which will output 0000 1111 on the 8 LED Output when the input pattern on the 8 Switch Input is 1111 0000. For all other switch inputs all the LEDs should be off.

   (a) Write a suitable flow chart or structured flow diagram.

   (b) Alongside it write a program for your controller.

4. What decimal or hexadecimal number is represented by the binary number 0001 0110?

5. Set up your controller with the 8 LED Output connected to the output. Write down a short program which will switch on LEDs to output 0000 1110. Input it into your controller and, when it is working, **ask your teacher to check it.**

6. An analogue light sensor is connected to the A to D Converter which is in turn connected to the input of your controller. The 2 Digit Display is connected to the Output of the controller.

In darkness the output of the A to D converter will be less than 0100 0000(binary) and in bright light it will be more than 1000 0000(binary).

   (a) Draw a flow chart or structured flow diagram showing how you would output 00 on the 2 Digit Display when the light sensor was in darkness, 99 when it was in bright light and 55 when the light was between these levels. (Remember, the 2 Digit Display works in binary coded decimal.)

   (b) Alongside this, write down a suitable program for your controller.

7. What is the main property of a **digital** quantity?

8. List 5 systems in which microprocessors are widely used.

9. What senses provide feedback for you?

Practical

QA Select a suitable system for question 3. Assemble the system. Program your controller and, when it is working, **ask your teacher to check it.**

QB Set the binary number 0001 0110 on the 8 Switch Input and program your controller so that it displays the switch settings on the 8 LED Output. **Ask your teacher to check it.**

QC Test your program for Q6. **Ask your teacher to check it.**

ASSESSMENT MODEL D

1. A radio-controlled boat has **three** radio channels, each of which can be ON or OFF. Channel 1 controls the motor which drives the propeller - ON turns the motor on and OFF turns it off.

Channels 2 and 3 control the rudder. When channels 2 and 3 are both OFF, the rudder is straight. If channel 2 is ON and channel 3 is OFF the rudder turns left. If channel 3 is ON and channel 2 is OFF, the rudder turns right.

Copy the table below and fill in the two right hand columns showing what happens to the motor and the rudder.

Channel 1	2	3	Motor for Propeller	Rudder
OFF	OFF	OFF		
ON	OFF	OFF		
ON	ON	OFF		
ON	OFF	ON		

If channels 1, 2 and 3 are connected to bits 0, 1 and 2 respectively of the output from a controller, what would happen when the controller outputs the following patterns:

a) 0000 0101?

b) 0000 0010?

c) 0000 0001?

What bit patterns would you output in order to make the boat

d) turn left?

e) turn right?

f) stop?

2. Two of the three systems listed below could easily be controller by a microprocessor. Which are they?

a) a lift in a shop

b) a classroom full of pupils

c) a central heating system

Choose **one** of the systems given above and make a list of all the sensors you would need.

Here is a list of possible sensors:-

Switches
Temperature sensor
Light sensor
Rotary potentiometer
Time switch
Reflective opto-switch
Liquid sensor
Slotted opto-switch

Make a list of suitable output devices for the **same** control system from this list:-

Lamp
Pump
Music Module
LEDs
Heater
Motor

3. Describe briefly 3 examples of microprocessor control systems which might be found in

a) your home

b) a factory

c) a hospital

(Give one example for each place.)

In each case, say what the microprocessor is controlling, what sensors it might need and what the microprocessor has to do.

4. What does a D to A Converter do?

Which of the following would need a D to A converter in order to get the best results?

a) an electric train

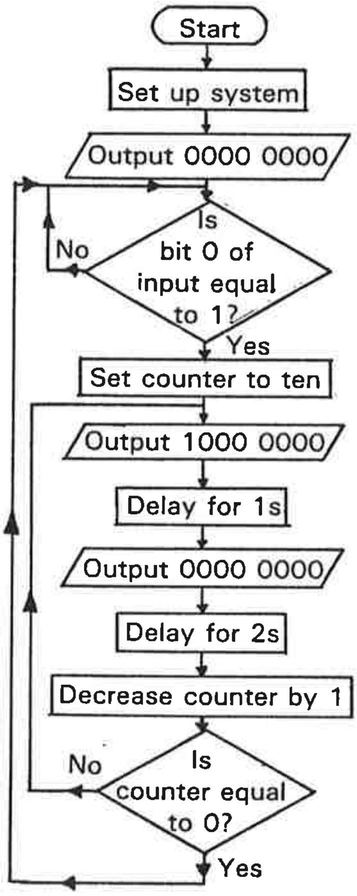
b) a traffic light

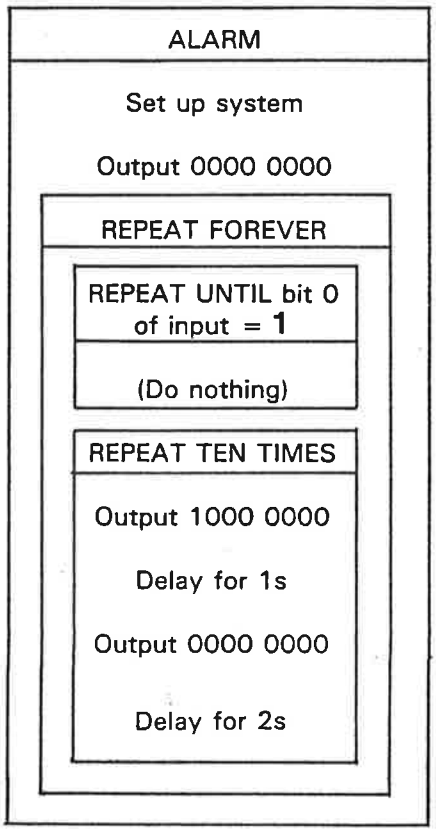
c) an alarm bell

d) a stage lighting unit

5. An alarm system is required to sound a buzzer ten times (on and off) if a push switch is pressed. The switch inputs a "1" into bit 0 of the controller input when it is pushed. The buzzer sounds when bit 7 of the controller output is set to 1.

Here is a suitable flow chart or structured flow diagram and, along side it, a program for the controller you used:-





a) Write down the program line(s) for your controller which

(i) Output 1000 0000

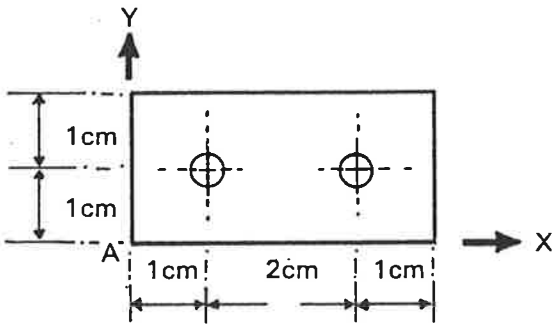
(ii) Repeatedly check bit 0 of the input until it is 1.

b) Write out a new flow chart or structured flow diagram which will perform the following control task with the **same** input and output systems:-

Sound a buzzer until the switch is pressed; when the switch is pressed turn the buzzer off for 5s.

c) Along side your flow chart or structured flow diagram write a suitable program for your controller.

6. An automatic drilling machine has to drill two holes in a metal plate in the position shown below



The drill is held in a frame which can move in steps of 1mm in either the X or the Y direction. There is a device to raise and lower the drill bit and the drill motor itself is controlled by the machine.

Initially the drill bit starts over position A. Describe precisely a suitable sequence of operations which specify how the machine can drill the two holes; the drill bit returning to position A again at the end.

7. Mrs Smith has a new greenhouse with electrically operated windows and heating system. Her plants are very delicate and if the temperature rises above 25 C the windows should open. On the other hand, if it falls below 10 C, the heating should come on. In addition to a temperature sensor there is one sensor which indicates when the windows are fully open and another to indicate when they are shut.

Using a flow chart or a structured flow diagram, explain how you would program a microprocessor controller to control Mrs Smith's greenhouse.

8. If someone suggested controlling these three systems with microprocessors:-

a) the London Underground

b) a heart transplant operation

c) the production and service of food in a restaurant

write down in each case, whether you think it would or would not be possible and desirable for microprocessors to be in control and explain your reasons.

10. Coding sheets and branch calculator

Coding Sheets

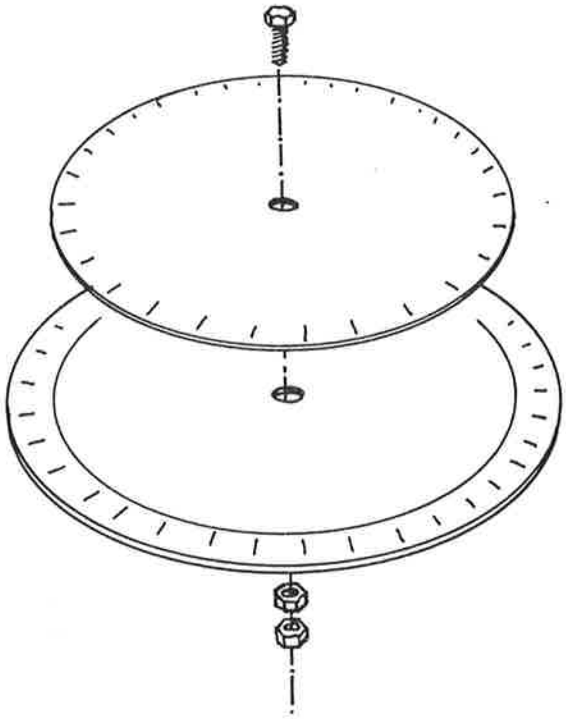
These may be copied for use by pupils developing programs for 3 Chip Plus.

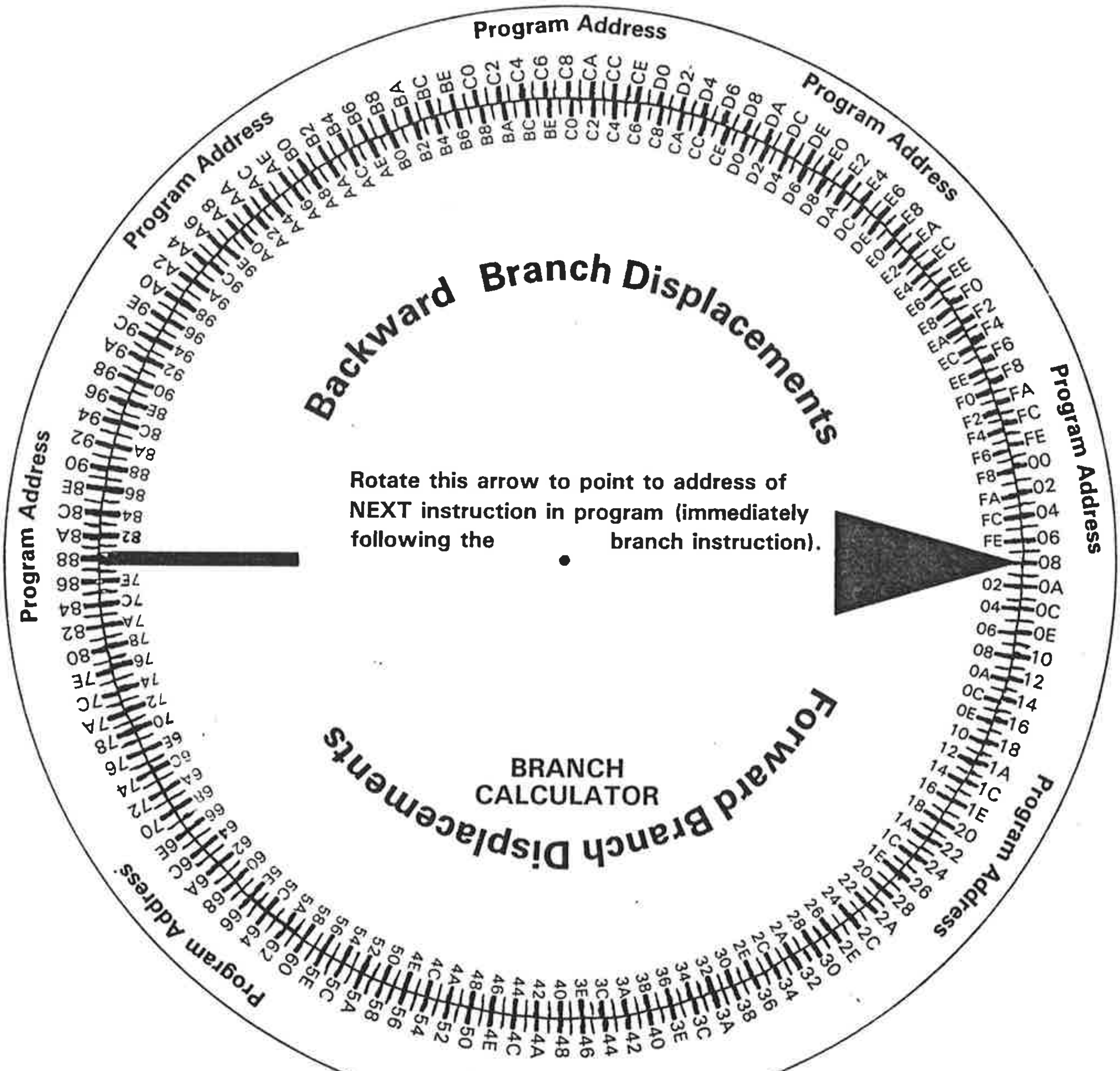
Address	Machine Code	Label	Mnemonics	Comments

Address	Machine Code	Label	Mnemonics	Comments

Branch Calculator

This is for use with 3-Chip Plus and should be constructed by copying the two circles and sticking them to thin card as shown.




Program Address
Backward Branch Displacements
Rotate this arrow to point to address of NEXT instruction in program (immediately following the branch instruction).
Forward Branch Displacements
BRANCH CALCULATOR