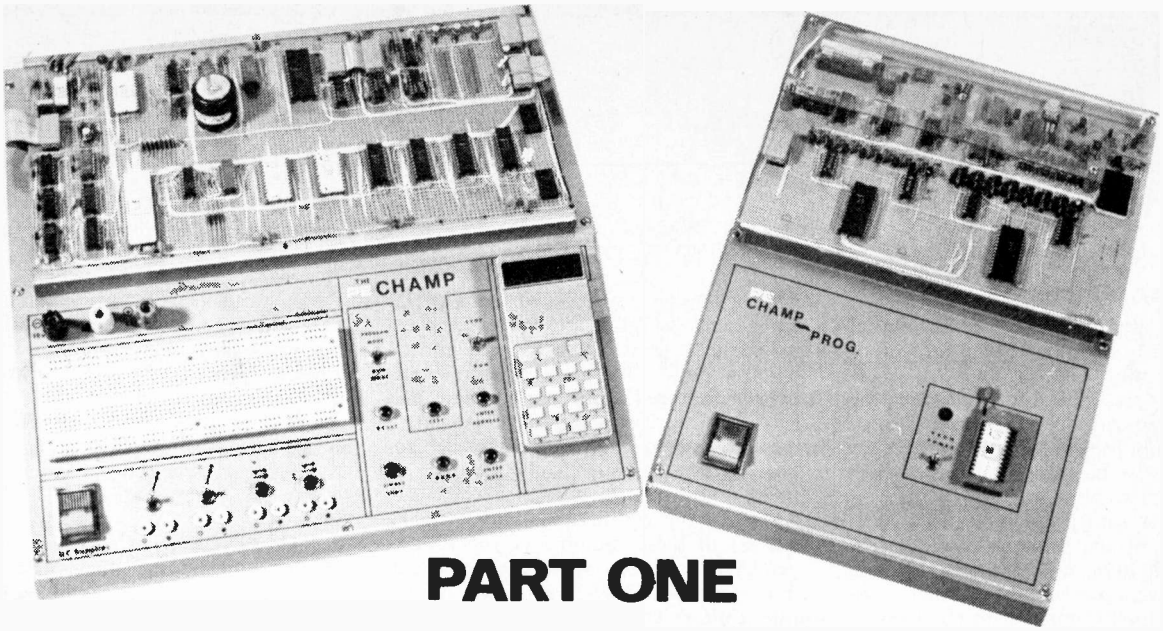


PART ONE

PE CHAMP

R.W.COLES
B.CULLEN

THE PE CHAMP PHILOSOPHY

THE microprocessor revolution is with us, and before long it will have steamrollered its way through to all but the most rudimentary electronic project areas. Before becoming a revolutionary, however, you have to make a careful choice concerning just what you want from this exciting new technology. A baby computer? A tutorial system? Or perhaps the ability to put together more effective logic and control systems? This choice is necessary because it is a fact that no reasonably priced microprocessor system can give you all of these things in a satisfactory combination.

CHAMP, in keeping with our "practical" image, is about making microprocessors work for a living; it is not a primitive attempt to emulate a full size data processing system which on grounds of cost alone, would fall short of desired performance in most respects. CHAMP is a development system which can be built economically without any requirement for expensive gadgets such as Teletypes, VDUs, or floppy discs; all you need can be built by you, including if you so wish, a PROM programmer and eraser.

WHAT CAN YOU DO WITH IT?

Well, you can't expect to sit down and write BASIC programs which beat you at chess. CHAMP is for examining switches, operating l.e.d.s and relays or playing "God Save the Queen", Stylophone style. It will control your model train layout or your f.m. tuner and Hi Fi system and later help you to put together small, cheap dedicated microprocessor boards to run the firmware programs you have developed to do these jobs. It has a calculator-type organisation which makes it easy for you to build up a collection of programs to crunch decimal numbers, if that is your interest. If you want to learn how to do practical things with microprocessors it will also form an excellent tutorial system to get you started.

HOW DOES IT DO IT?

Well, cheaply for a start. CHAMP uses one of the cheapest microprocessor chips around, but nevertheless has plenty of capability which you would be hard pressed to employ to the full. In our opinion it's no use investing in the fast, all-bells-and-whistles MPU chips unless you can afford the memory and peripheral facilities required to take advantage of their power.

With CHAMP you can write your programs into RAM while you are developing them, but you won't lose them when you hit the mains-off switch. CHAMP program RAM, all 512 eight-bit words of it, is of the CMOS variety which is supported by batteries when the power is off. This advanced feature removes the need for paper-tape readers or cassette interfaces and puts you streets ahead of some of the "noddy" development kits now on the market.

You write your programs into RAM with the aid of a simple hexadecimal keyboard and display, under the supervision of routines in the CHOMP firmware which (to save you buying a lot of expensive PROMs) fits into a single 1702A device.

Interfacing CHAMP is simple because a low cost, program-configured interface chip is provided on the board to allow you to connect up to just about anything from a psychedelic light show to an array of toggle switches. The only addition required is the appropriate buffering, which can be mounted on the integral bread-board socket strip provided for this purpose.

In a nutshell then, the CHAMP philosophy concerns putting microprocessors to work in practical situations as cheaply as possible, using familiar constructional techniques to build a completely self-contained system with simple operating procedures.

SYSTEM FEATURES

★*Calculator-based hexadecimal keyboard and display allow rapid program entry and debug*

★*On-board battery supported CMOS RAM for up to 512 program steps allows retention of carefully entered programs with power off*

★*Based on well supported Intel 4040 microprocessor. Has 60 easy to learn instructions which include binary and decimal arithmetic and keyboard encoding*

★*Twenty-four on-chip registers and interrupt facility available*

★*System monitor program CHOMP allows examination or modification of program memory, and program execution in single-step mode*

★*Open plan design allows easy access to all parts of circuit; no expensive plug-in cards to buy*

★*Programmable input/output interface chip makes for versatile interfacing at the breadboard level*

★*Optional PROM programmer and eraser allows a full development cycle to be undertaken by the home constructor for the very first time*

★*No need for any expensive peripheral equipment like TTY or VDU*

1—MEET THE FAMILY

CHAMP stands for low *C*ost *H*ighly *A*daptable *M*icro *P*rocessor, and we feel that those words really do sum up the main attributes of this versatile and capable system. CHAMP is *not* a toy for you to merely sample the wonders of *real* computers. CHAMP *is* a real computer in its own class, and it will not leave you wishing that you could afford a floppy disc system or another 8K of RAM, because it does not require these things to operate effectively. CHAMP is a development system for both software *and* hardware which will enable you to develop functional programs for their own sake, or for later inclusion in small minimal-hardware systems which could be built for a very small outlay.

CHAMP does not live on a collection of very expensive plug-in cards which have to go into an equally expensive 19in rack case (where you can't get at them anyway without expensive extender cards). Instead it fits on one large "open-plan" 0·1in matrix Veroboard which costs a lot less than a *single* plug in DIP board and allows you to stick your multimeter probes anywhere you like without being a contortionist.

There is ample room on the board for you to add ideas of your own, and all the buses to which you may need access are brought out on the cheap 16-pin DIL sockets so that you can debug or expand your system easily. The CHAMP circuit board slides into guides mounted above an attractive low profile plinth which houses the power supplies and also provides the control panel and a hardware development area.

KEYBOARD

The only peripheral equipment you need to enter and examine your programs is a simple hexadecimal keyboard and display unit which uses 16 keys labelled 0–9 and A–F and can display eight digits in seven-segment format. One of

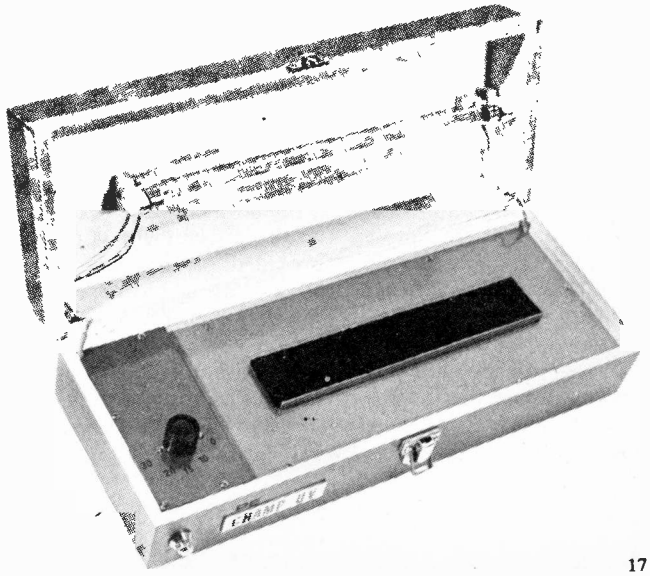
Estimated Component Costs...

CHAMP
with 256 program bytes, power supply, keyboard and breadboard socket £120

CHAMP—PROG plus extra PROM £60

CHAMP—UV £12

Full list of i.c.s required will appear next month



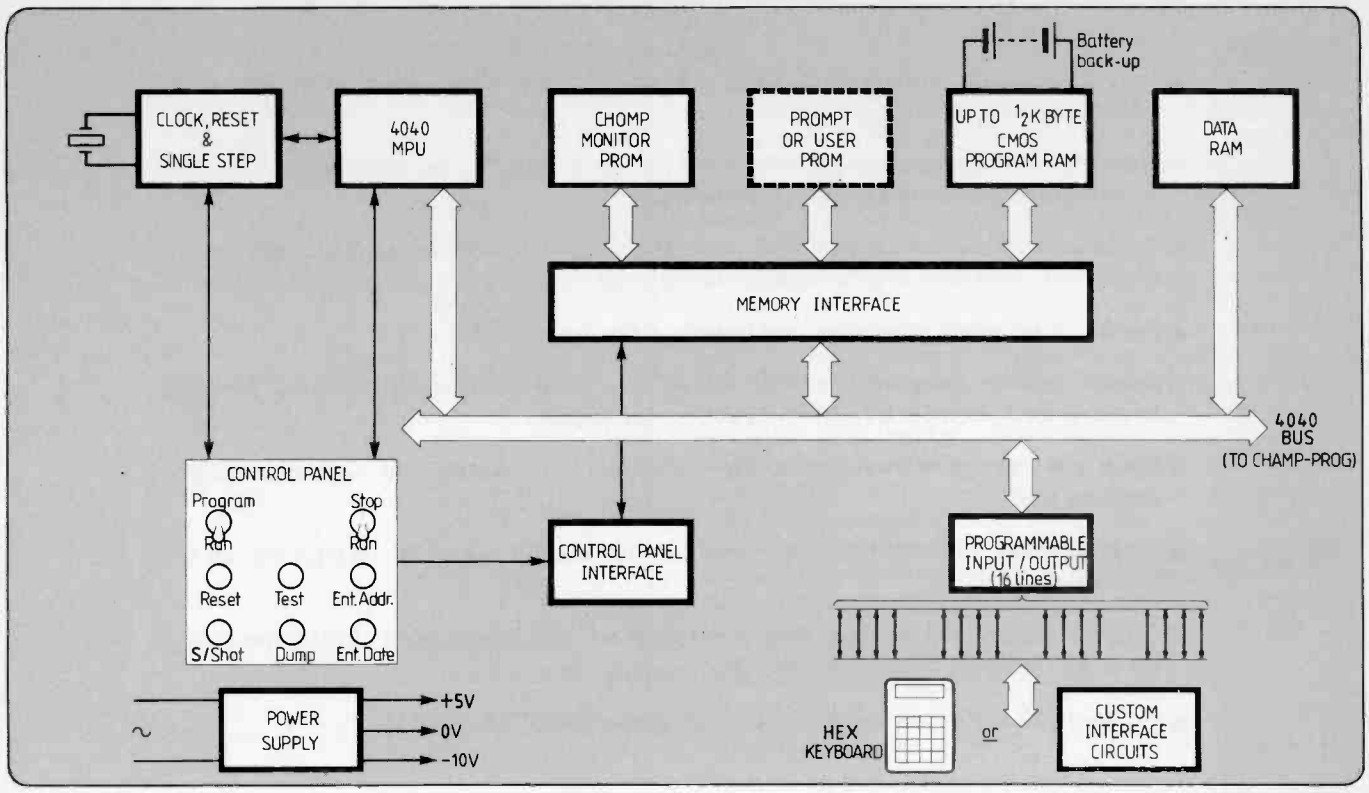


Fig. 1.1. Basic block diagram of the PE CHAMP Development System

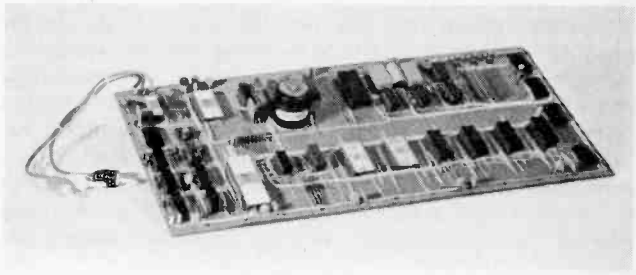
the simplest and cheapest ways to build this keyboard is to modify one of the now commonplace "throw-away" calculators, and constructional details for this and a "build from scratch" version will be provided later.

ENTERING AND RUNNING PROGRAMS

Before you can enter programs of course, you have to write them, and this is done using the evolutionary process of vague flow-chart to detailed flow-chart to list of hexadecimal coded instructions, as developed in Part 3 of *Microprocessors Explained*. Examples of this procedure will be given later, together with some sample programs for you to try out.

When power is applied CHAMP is reset and the display shows the hexadecimal equivalent of the first program RAM address. Pressing keys on the keyboard will enter hexadecimal code up to three digits long and this can be entered either as a new address (by pressing ENTER ADDRESS) or as a program instruction/data value (by pressing ENTER DATA). After the ENTER DATA switch is pressed, the current address is incremented by one to point to the next available location in RAM, ready for further data. If you like, you can examine the existing contents of a RAM location using the DUMP switch whereupon the address will again be incremented to allow you to step through a program and compare it with your paper version. A RESET switch is provided so that any current operation can be aborted, and RUN/STOP and SINGLE SHOT switches allow you to run a program one instruction at a time for debugging purposes.

When you are satisfied with a program you have entered you can throw the MODE switch to the RUN position to allow your program to do its stuff, exiting if necessary with the RESET switch.



CHAMP circuit board

MPU CHIP

CHAMP is built around an INTEL 4040 microprocessor, but provides facilities for the development of programs and hardware for *either* the 4040 itself *or* the simpler 4004. This dual processor versatility is possible because 4004 instructions form a subset of the 4040 instruction set, making it possible to run any program written for the 4004 on a 4040 system. All that you have to remember when writing 4004 programs to try out on CHAMP is that you must not include any "4040 only" instructions in your code.

The 4040 chip on CHAMP itself forms part of a complete microprocessor system which includes a versatile crystal clock circuit and both program and data memory. Two input/output interfaces are provided, one dedicated to the control panel switches and one which can be used either by the hexadecimal keyboard or by any external hardware provided by the user. A single level interrupt is provided which is used by the keyboard but which can also be used externally, when required for any particular application.

MEMORY

With the 4040 system organisation, data RAM is kept separate from program RAM, leading to an easy to understand, calculator-type architecture which is extremely easy to use.

Eight kilobits of program storage are available on a fully stuffed CHAMP board, organised as 1024 eight-bit words. Of these, 512 words are available as two 1702 u.v.-erasable PROM chips, and a further 512 words are available as four 5101 CMOS RAM chips. Only one PROM chip is required for the CHOMP firmware, although both chips will be required if the PROM programmer unit is added later. The CMOS RAM can be added in two 256-word increments if desired, allowing a minimum operational CHAMP configuration of CHOMP (256 words) + 256 words of RAM available for user programs. The CMOS RAMS have a very low current drain when on standby, and because of this they can be made to retain their data without mains power with the aid of a simple power-fail-detect circuit and a 3-cell DEAC battery. This part of the CHAMP circuitry is unique, and gets over that awful problem experienced with most development systems where you have to lose all your laboriously entered hexadecimal code at the end of the day, a problem which usually forces users to buy paper-tape systems or cassette interfaces to allow rapid program re-entry at the start of a new session.

CHOMP

CHOMP stands for *CH*amp *O*perating system and *M*onitor *P*rogram and is the name given to the collection of programs required to control all the CHAMP facilities during program entry and debug. CHOMP is entered automatically at power on, and can always be restarted at any time by pressing the RESET button. CHOMP contains routines for refreshing the eight-digit seven-segment display, for accepting and storing keyboard entries, for entering instructions or data into RAM, for examining existing RAM content and for entering a new effective address.

In order to keep costs down, CHOMP has been made small enough to fit into a single 1702A PROM and has been written so as to provide a range of programming examples for those new to the game. Later in this series the complete listing of CHOMP will be provided, together with a full description of how it works. Arrangements have been made for the supply of ready programmed PROMS and also for CHOMP to be loaded into constructors' existing PROMS if required, full details of this service will be provided later.

CHAMP-PROG

CHAMP-PROG is an optional PROM programming accessory for CHAMP which extends the system's usefulness immeasurably. Using this unit, programs developed and debugged in RAM can be transferred to permanent storage in the form of low-cost 1702A PROMS which can be erased and reprogrammed as often as required.

This facility turns CHAMP into a no-compromise development system which enables you to:

(i) Develop and debug programs in CHAMP RAM.
(ii) Develop and debug interface hardware on the CHAMP breadboard, and with CHAMP-PROG.
(iii) Dump your working programs into PROM chips and plug these into "Sons-of-CHAMP" minimal microprocessor circuits which can then be used to carry out dedicated tasks.

This classic use of PROM programming is not the *only* way in which CHAMP-PROG extends CHAMP facilities however; PROMS can now be used simply for the long term storage

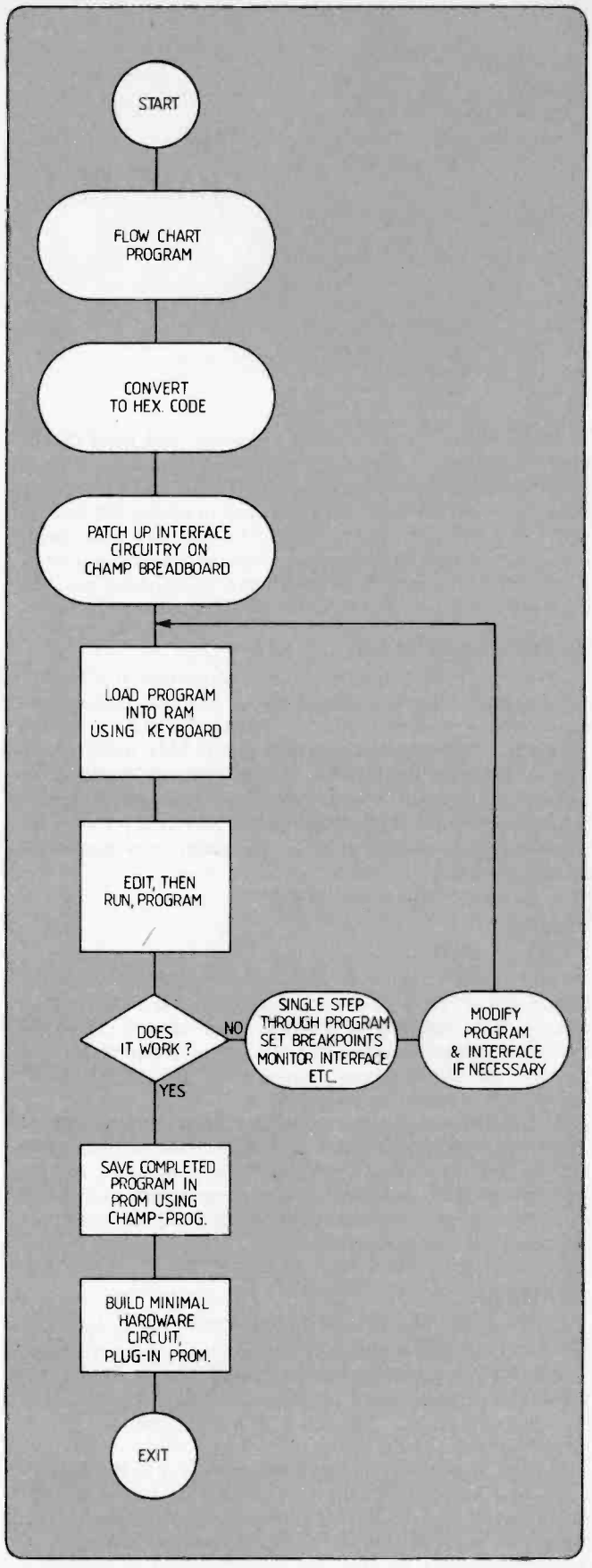
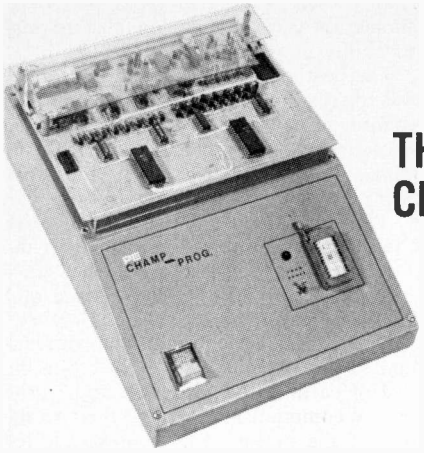


Fig. 1.2. Using CHAMP as a development system



THE CHAMP-PROG

of useful but no longer current programs, and these can be kept in a "box-on-the-shelf" until next required. This is possible because not only does CHAMP-PROG let you dump RAM into PROM, it also allows you to reverse the process and dump PROM back into RAM. In this way a firmware library can be built up, consisting of useful program segments, games programs, etc. all of which can be reloaded into the CHAMP operational RAM area in just a few seconds.

MATCHING PLINTH

CHAMP-PROG is designed to be a companion to CHAMP in every way, including the use of an identical layout technique and a matching plinth. CHAMP-PROG uses the +5V and −10V supplies provided by CHAMP itself, and in addition contains the 80V d.c. supply required for the programming operation. Areas of the "open-plan" circuit board carrying this higher voltage are screened off with the aid of a clear plastic window to prevent any accidental catastrophes with bare wires.

PROMPT

CHAMP-PROG is controlled by a firmware program called PROMPT (PROM *P*rogramming *T*echnique) which fits into a 1702A PROM and is plugged into the second PROM socket on CHAMP. PROMPT uses several CHOMP routines and also uses the CHAMP keyboard and display for the entry of programming requirements and the display of "next entry" prompting messages.

To program a PROM the required address range is entered via the keyboard, the PROM is inserted into the zero insertion force socket, the PROGRAM POWER switch is turned on, and the PROGRAM button depressed. The programming of a full PROM takes about two minutes although dumping PROM data into RAM is of course much faster.

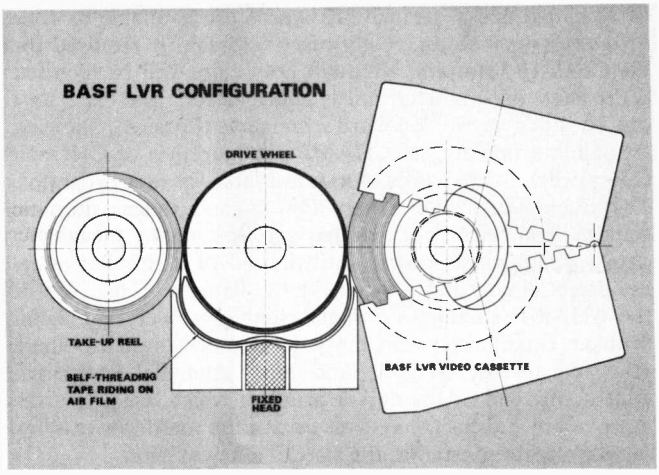
CHAMP-UV

CHAMP-UV, the last of the CHAMP family, is a *safe* but low cost PROM eraser which becomes necessary when CHAMP-PROG is added to the CHAMP system. CHAMP-UV will erase in a matter of minutes any u.v.-erasable PROM including the 1702A, 2704 and 2708 variety, and can handle up to 10 of these at a time.

Erasure is achieved in a light-tight box which houses a short-wave u.v. tube interlocked so that it is always "off" when the lid is open. This interlock is necessary because short-wave u.v. light can be harmful to the eyes and skin. A timer is included so that the over-exposure of expensive chips can be avoided.

NEXT MONTH: The PE Champ Circuitry

NEWS BRIEFS



Low Cost Video Recorder

An agreement recently signed between BASF AG and Bell & Howell Corporation is expected to result in the development for the amateur market of a video cassette recorder based on BASF's Linear Video Recording (LVR) technique. The new recorder is expected to be on the market in time for Christmas 1979.

The LVR cassettes are claimed to occupy only about one quarter of the volume of other video cassettes, yet provide double the playing time. The 6·25mm CrO_2 tape travels backwards and forwards past a stationary head at 3m/sec, and is recorded with 28 tracks.

Economical, high-quality tape duplication is possible in a single pass, using a multiple head.

The small size of the cassette, the low tape consumption, and the use of a simple mechanism and sophisticated electronics mean that the LVR system offers great potential for miniaturisation. For example the tape transport system could be integrated into a video camera.

AVM For Dublin Buses

The Dublin City Bus Service's entire fleet is to have what is claimed to be the world's most advanced automatic vehicle monitoring system (AVM), supplied mainly by Storno Ltd. of Camberley, Surrey.

Each bus will be fitted with an odometer, the initial reading of which is fed to a computer at the outset of each journey. Control Inspectors from seven different garages will be able to locate precisely, any bus, using telephone line modems feeding VDU terminals.

A Storno CQF612 v.h.f. Data Transmitter situated on Three Rock Mountain, will be u.h.f. linked to the computer at O'Connell Street. The bus-borne control heads are also capable of speech communication directly to the garage, if requested.

In addition, the Storno bus data unit stores the bus location *and* passenger loading, as well as up to ten mechanical parameters such as engine temperature and oil pressure. The computer system will be interrogating for data at the rate of 900 buses per minute, using a fully duplicated radio system.

Control Inspectors will also be able to watch traffic and passenger queue conditions using remotely controllable CCTV cameras at city centre points.

The system is expected to be in operation by mid 1979.