PE CHAMP

R.W. COLES B. CULLEN

PART TEN

CHAMP—U.V.

THE combination of CHAMP and CHAMP-PROG produces a microprocessor development system which can be used not only to experiment with software techniques, but also to produce other working microprocessor systems with dedicated PROM based programs and hardware interface circuitry which has been fully tested in advance on the CHAMP breadboard.

This month we conclude the CHAMP series with a deeper examination of the uses to which CHAMP-PROG can be put, and with the constructional details of the CHAMP-U.V. PROM erasing system.

USING CHAMP-PROG

You may remember that in part one of the CHAMP series we stated that CHAMP-PROG extended the CHAMP system not only in its role as a PROM programmer but also as a system for copying data *already* stored in a PROM *back* into the CHAMP program RAM. A facility for copying data into RAM makes the whole system even more flexible and allows the following:

(i) Programs can be "dumped" into a PROM to release the valuable CHAMP CMOS RAM area for more pressing jobs with the knowledge that the original programs can be easily reloaded into RAM when necessary.

(ii) PROMS can be modified by loading their contents into CHAMP RAM, making the necessary changes, and then reloading the erased PROM with the updated contents.

(iii) PROMS can be "insert/delete" edited using the relocating ability of the PROMPT program.

(iv) PROMS can be duplicated by first copying them into RAM and then using CHAMP-PROG in the usual way.

The PROMPT software published last month does not have the ability at present to load data back into RAM, because there simply was not room in the single 4702A chip to do this. The PROMPT program is, however, laid out in such a way that it is very easy to produce a new version, say PROMPT II, which *will* perform this useful function. Before you start to groan at the prospect of *another* financial outlay for software, remember that if you already have a CHAMP-PROG and PROMPT, then PROMPT II will only cost you the price of an additional PROM because of course you can now do all the programming yourself! In addition, the generation of PROMPT II makes an excellent introduction to the use of the CHAMP-PROG system, and so we have treated the production of this new firmware as a worked example accordingly.

WUNBYTE II

If you examine the flowchart and listing of PROMPT published last month you will notice that the only part of the program directly involved with the programming hardware is the subroutine WUNBYTE which addresses the source and destination, and generates with software the accurately timed program-enable pulse. To generate the new PROMPT II firmware all that is required is the replacement of this subroutine by a new one which transfers data in the other direction. The new subroutine can be called WUNBYTE II and does not need to be as lengthy or as complex as the original because there is no longer a need for the program-enable timing counter. A listing of the new subroutine is shown in Fig. 10.1. and as you can see WUNBYTE II starts at the same address as WUNBYTE so that the JMS WUNBYTE call is still effective. The new subroutine is shorter than the original, but this is of no consequence because the BBL instruction will pop the stack as usual, and operations will recommence at the line immediately following the JMS.

Apart from the new subroutine and an area of blank space following it, the rest of PROMPT II is identical to the original PROMPT, and of course the new PROM is destined for the same socket on the CHAMP main board, where it can be used alternately with the original when required.

PROGRAMMING SEQUENCE

To create PROMPT II the following sequence must be followed.

(i) Connect up CHAMP-PROG to CHAMP and connect the mains supplies. Ensure that CHOMP and PROMPT are in their respective sockets on the CHAMP main board, and place an erased PROM in the CHAMP-PROG programming socket.

(ii) Switch on CHAMP and enter the 56 lines of WUNBYTE II into CHAMP program RAM starting at, say, address 200H (you could of course start *anywhere* in the CHAMP RAM space).

(iii) Press RESET then TEST to enter PROMPT, turn on the CHAMP-PROG mains, and then enter Adr1, Adr2 and Adr3 as follows to copy WUNBYTE II into the new PROM at location 15AH.

Adr1 = 200H (for example)
Adr2 = 237H (200H + 56 decimal)
Adr3 = 05AH (destination in PROM)

(Remember to turn the PROM POWER switch to the ON position immediately before pressing the ENTER DATA button after Adr3 has been keyed in).

MCS 40 PROGRAM SHEET

TITLE	WUNBYTE II (ALTERNATIVE TO WUNBYTE FOR PROMPT)	DATE 09 01 78	PAGE NO. 1 OF 3

PAGE	LINE	ROM CODING	LABEL	OPERATION	OPERAND	COMMENTS
1	4 0					*THIS SUBROUTINE
	1					CAN BE USED IN
	2					PROMPT INSTEAD OF
	3					WUNBYTE & HAS
	4					THE EFFECT OF
	5					LOADING SOURCE
	6					DATA IN PROM BACK
	7					INTO CHAMP
	8					PROGRAM RAM*
	9					
	A					Adr 1 = Start of source
	B					in P.ROM.
	C					Adr 2 = End of source.
	D					
	E					Adr 3 = Start of destination
	F					in R.A.M.
	0					
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	8					
1	5 9					
1	5 A	D1	WUNBYTE	LDM	1	PROM → RAM S.ROUTINE
	B	FD		DCL		ADDRESS 4265 No 1
	C	28		FIM	8	
	D	80		8	0	
	E	29		SRC	9	LOAD PORTS W & X
	F	A3		LD	3	WITH SOURCE ADDRESS

ORIGINAL PROMPT

MCS 40 PROGRAM SHEET

TITLE	WUNBYTE II	DATE 09 01 78	PAGE NO. 2 OF 3

PAGE	LINE	ROM CODING	LABEL	OPERATION	OPERAND	COMMENTS
1	6 0	E4		WR0		IN P.R.O.M.
	1	A2		LD	2	
	2	E5		WR1		
	3	D2		LDM	2	ADDRESS 4265 No 2
	4	FD		DCL		
	5	29		SRC	9	
	6	0B		SB1		SELECT REG. BANK 1
	7	EC		RD0		READ PROM (1st NIBBLE)
	8	F4		CMA		COMPLEMENT IT.
	9	B1		XCH	1	PUT IN REG 1
	A	ED		RD1		READ PROM (2nd NIBBLE)
	B	F4		CMA		COMPLEMENT IT
	C	B0		XCH	0	PUT IN REG 0.
	D	D0		LDM	0	
	E	FD		DCL		
	F	28		FIM	8	SELECT DATA RAM
1	7 0	10		1	0	CHIP 0 REG 1, CHAR 0
	1	29		SRC	9	
	2	A1		LD	1	
	3	E0		WRM		STORE 1st NIBBLE FOR MATCH.
	4	69		INC	9	NEXT RAM CHAR.
	5	29		SRC	9	
	6	A0		LD	0	STORE 2ND NIBBLE FOR MATCH.
	7	E0		WRM		
	8	28		FIM	8	ADDRESS CORRECT
	9	00		0	0	PROGRAM RAM
	A	29		SRC	9	DESTINATION CHIP.
	B	AE		LD	E	
	C	E1		WMP		
	D	2D		SRC	D	SELECT PROGRAM RAM
	E	A1		LD	1	BYTE
	F	E3		WPM		WRITE 1st NIBBLE

MCS 4C PROGRAM SHEET

TITLE	WUNBYTE II	DATE 09 01 78	PAGE NO. 3 OF 3

PAGE	LINE	ROM CODING	LABEL	OPERATION	OPERAND	COMMENTS
1	8 0	A0		LD	0	
	1	E3		NPM		WRITE 2ND NIBBLE
	2	00		NOP		
	3	00		NOP		
	4	0E		RPM		READ DATA BACK
	5	B5		XCH	5	FOR MATCH CHECK
	6	0E		RPM		
	7	B4		XCH	4	
	8	28		FIM	8	
	9	20		2	0	
	A	29		SRC	9	STORE MATCH DATA
	B	A5		LD	5	IN DATA RAM
	C	E0		WRM		
	D	69		INC	9	
	E	29		SRC	9	
	F	A4		LD	4	
1	9 0	E0		WRM		
1	9 1	C0		BBL	0	END OF NUNBYTE II
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					

NOPS TO 1A5 THEN PROMPT CONTINUES WITH MATCH

Fig. 10.1. WUNBYTE II program listing

(iv) When the "done" message is displayed, turn the PROM POWER switch to the OFF position and then press RESET and TEST to re-enter PROMPT. Enter the following addresses to copy the part of PROMPT *before* WUNBYTE into the new PROM, and then initiate programming in the usual way.

Adr1 = 100H (start of PROMPT)
Adr2 = 159H
Adr3 = 000H (start of PROMPT II)

(v) When "done" is displayed once more, turn the PROM POWER switch to the OFF position, and then press RESET and TEST to re-enter PROMPT. The following addresses are then entered to program the remaining part of PROMPT into the new PROM in the usual way.

Adr1 = 1A6H (start of MATCH)
Adr2 = 1FFH (end of PROMPT)
Adr3 = 0A6H

(vi) When "done is again displayed turn off the PROM power and remove the newly programmed PROM. PROMPT II now lives!

USING PROMPT II

Apart from the subroutine WUNBYTE, the new PROMPT II is identical to the old one, and addresses are entered exactly as before with the same meanings:

Adr1 = start of source data
Adr2 = end of source data
Adr3 = start of destination area

The important difference is of course that the *source* addresses now refer to a PROM in the programming socket, and the *destination* address refers to a location in CHAMP program RAM. This also means that the most significant digit of addresses 1 and 2 is redundant and can be set to anything (usually zero to avoid confusion), and that the most significant digit of Adr3 now becomes important and is used to select the destination chip in CHAMP program RAM. Provided you think in terms of source and destination rather than RAM and PROM no confusion should arise when swopping from one PROMPT version to the other! Data transfer is initiated as usual by the depression of ENTER DATA after keying in Adr3, although now the "done" message is displayed almost immediately because the transfer to RAM is very much faster. When using WUNBYTE II, there is no need to turn the PROGRAM POWER switch to ON of course, since all the necessary power is supplied by CHAMP itself.

OTHER POSSIBILITIES

No doubt many readers have spotted the fact that with extra PROM space available on CHAMP, the two versions of PROMPT could both be available simultaneously, and of course there would be no need to duplicate all of the original PROMPT, only the WUNBYTE II would be needed, with some means of selecting *which* subroutine is used by PROMPT when it is run. (This would require a reference to switches, and the ENTER DATA and ENTER ADDRESS buttons could easily be redefined for this purpose.)

Of course the addition of an extra PROM or PROMS is a major modification to CHAMP, but this could nevertheless be done without too much trouble.

If you *did* add this extra PROM it would of course be mostly empty, and then your thoughts could turn to what *other* goodies you could add to the system. How about a WUNBYTE III which would not *transfer* data but simply check one area against another, thus providing a new VERIFY facility in fact. Or how about a WUNBYTE IV which would be used to move data around in CHAMP RAM and would not need the benefit of CHAMP-PROG at all. Insert/delete editing would be possible with WUNBYTE IV providing some spare locations were left at the start of a program (think about it). Flow charts for these other WUNBYTE variants are shown in Fig. 10.2 and these could be the source of a lot of fun for CHAMP users, with many other possibilities no doubt suggesting themselves as experience is gained.

PROM ERASER

An essential companion to a PROM programmer like CHAMP-PROG is of course a PROM eraser, and fortunately these units are not difficult to build. All that's really required is a lightproof box with a short wave ultraviolet (U.V.) lamp mounted inside it and a resting place for the PROMS being erased. We decided against the minimal "lash up" approach to the design of an eraser at a very early stage for two very important reasons:

(i) Short wave U.V. light at the correct wavelength of 2537 Angstroms can be harmful to living organisms, and this of course includes CHAMP programmers!

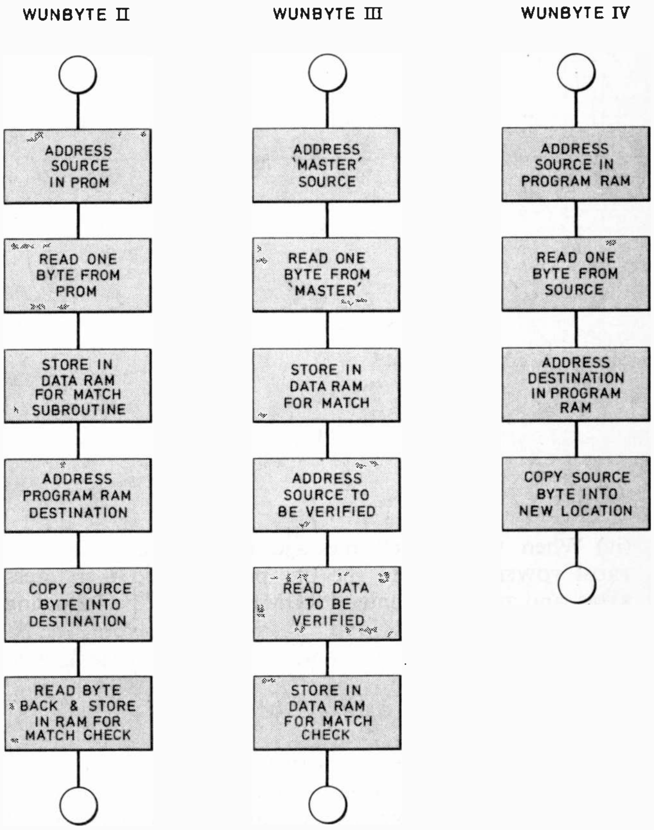
(ii) Over exposure of 4702A PROMS to the erase light can shorten their life.

In order to avoid the implications of both (i) and (ii), a good deal of thought was put in to make CHAMP-U.V. safe for both the programmer *and* PROM. Technically, there is no real problem to building a PROM eraser, short wave lamps are available from specialist companies such as Anderman who seemingly stock them primarily for medical purposes. The lamps resemble normal 8 watt fluorescent strip lights, and use exactly the same miniature 2-pin base, and the same ballast choke and starter. Lamp power is provided by the usual 240V a.c. mains supply. The required integrated light dose, which is defined as intensity × exposure time, is stated by Intel to be six Watt-Seconds per square centimetre, and this can be provided quite quickly by the small 8W tubes when they are placed within 1in of the PROM.

CHAMP U.V.

CHAMP-U.V. is built in a plywood case measuring 380 × 140 × 100mm and is completely self contained. Eye safety is ensured by the use of a microswitch which will not allow the lamp to light until the lid is closed and secured. PROM safety is ensured by the incorporation of a clockwork timeswitch which allows "set-and-forget" erasing to be undertaken.

Fig. 10.2. Flowcharts for WUNBYTEs II, III & IV. Note: WUNBYTE IV makes a RAM block-move possible. Source and destination blocks can overlay each other, but with an incrementing address counter data can only be moved *down* in RAM, if over-writing unused source data is to be avoided. Modifying PROMPT to permit address decrementing would be possible, and would allow data to be moved *up* (i.e. to a higher start address). WUNBYTE IV allows program blocks to be moved aside to make room for single extra instructions when required

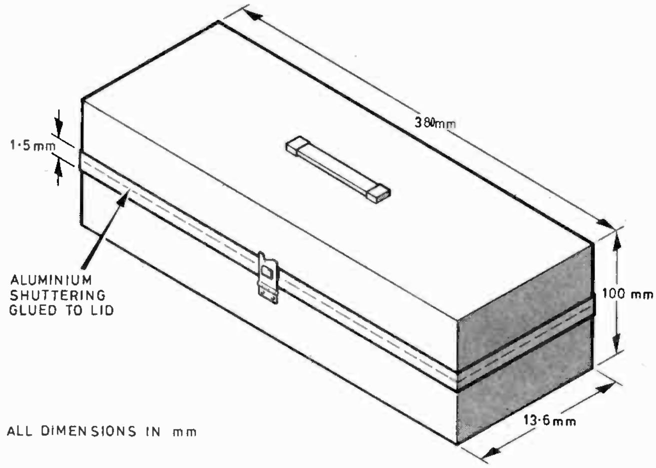


The circuit of CHAMP-U.V. is shown in Fig. 10.4, and as you can see, after the complexities of CHAMP and CHAMP-PROG, CHAMP-U.V. should come as a spot of light relief! The microswitch, which is operated by the box lid, is in series with the mains input, as is the clockwork timer. The neon mains indicator is wired to show that mains is applied correctly when the lid is closed, and not to show that the time-switch is still on. The timeswitch itself makes a distinct sound rather like the ticking of a time-bomb when active, and so there seemed little need for additional facilities to announce the end of an erase cycle. The timeswitch can be set by means of a knob inside the case to provide erase cycles ranging from zero to thirty minutes, with periods of about twenty minutes being the norm for 4702A devices.

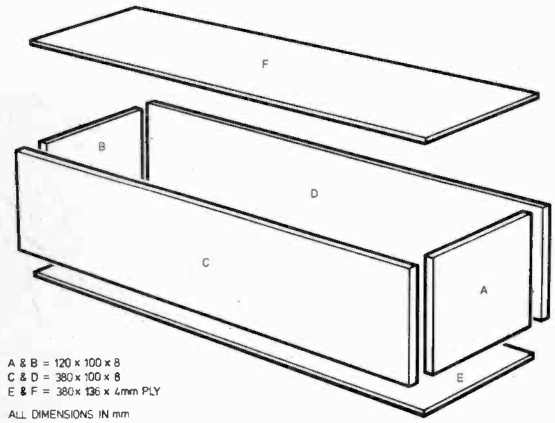
The lamp is mounted lengthways in the lid of the case and provides a full 200mm of active length for the erasure of PROMS. A total of twelve PROMS can be erased simultaneously when required, and these are mounted on a strip of conductive plastics foam which is located immediately under the tube when the lid is in the closed position.

CONSTRUCTION

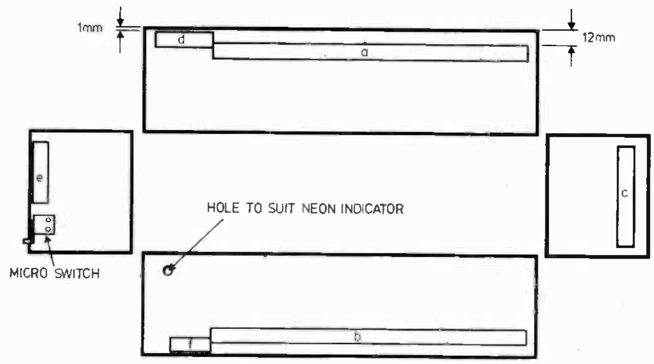
The CHAMP-U.V. case is made from 8mm plywood and is pinned and glued together using simple butt joints. The best way to build the case is to start by assembling the body and the lid as one piece and then to saw the resulting box completely through to separate the lid. Plywood runners are



(a) Basic case dimensions

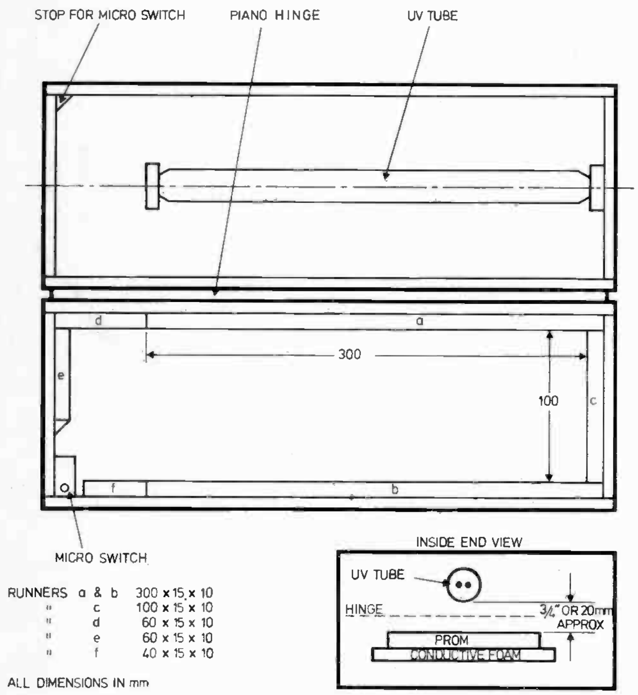


(b) Plywood parts

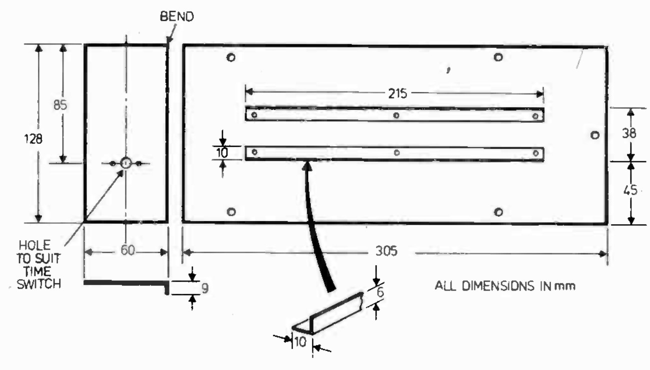


(c) Runner positions

Fig. 10.3. Constructional details of CHAMP-U.V.



(d) Internal layout of case



(e) Aluminium parts

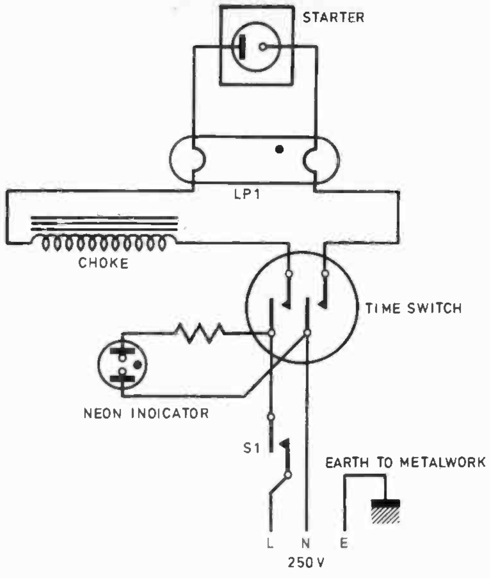


Fig. 10.4. Circuit diagram of CHAMP-U.V.

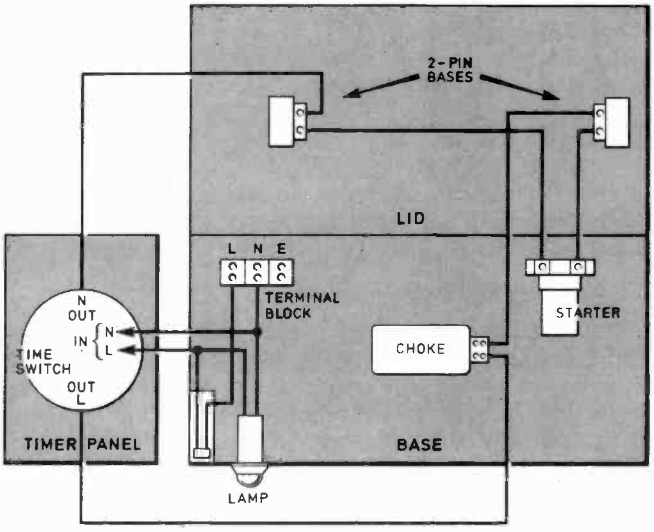


Fig. 10.5. Wiring layout of U.V. unit

It is most important that the lid operated microswitch feature of this design is incorporated, to immediately switch off the U.V. tube when the top is lifted. The optimum position for the microswitch is *near the front* of the box as shown in this photograph

COMPONENTS...

Ultraviolet tube. 12 inch, 8 watt, 2537 Angstrom

Choke. Smart & Brown 69386, 8 watt, 0·16A (or similar).

Starter. GEC 155/100 (or similar).

Timeswitch. 30 minute clockwork timer unit. Contacts *closed* during timing period, and of mains rating. (Available from many surplus suppliers).

Microswitch. 240V a.c., 2A, with leaf spring and actuating plunger.

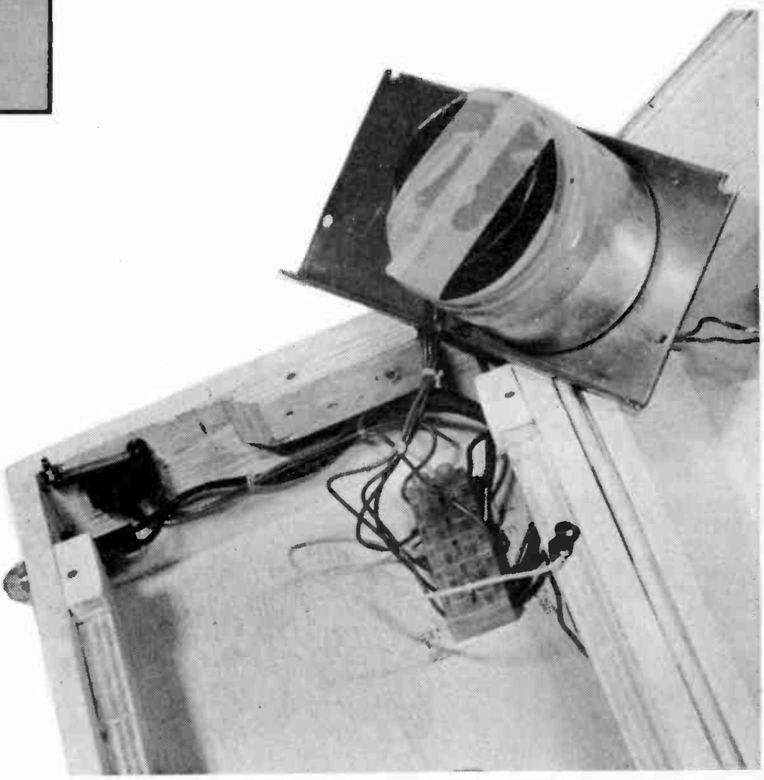
Neon indicator lamp. 240V type.

Miscellaneous. 2-pin tube sockets. 3-way terminal block, 4 mm and 8 mm plywood, aluminium sheets, etc.

The U.V. tube is available from Anderman & Co. Ltd., Central Avenue, East Molesey, Surrey KT8 0QZ.

WARNING

It is *essential* that these U.V. lamps are used correctly if health damage through U.V. exposure is to be avoided. Observe the manufacturers caution fully.



then pinned and glued inside the box to locate the timer panel and the PROM panel, which are both cut from aluminium sheet. Two aluminium brackets are also cut out and bent to form a PROM carrier which can be lined with conductive plastics foam when the rest of the case is completed (Fig. 10.3).

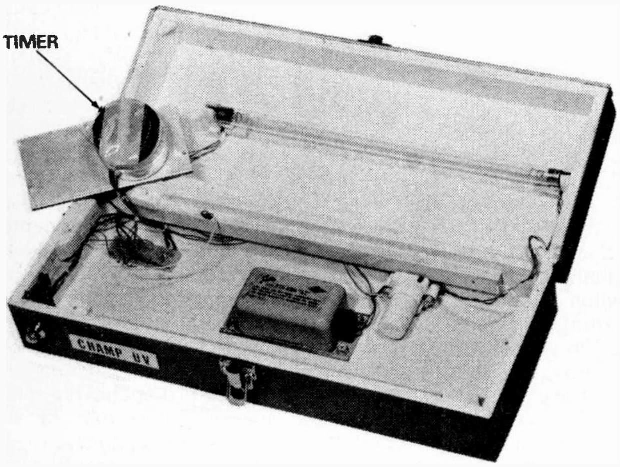
The lid is attached to the body of the case by means of a plastics piano hinge secured externally with small wood-screws, and this hinge, together with aluminium shuttering glued along the other three edges of the lid, ensures a light-tight seal for complete safety.

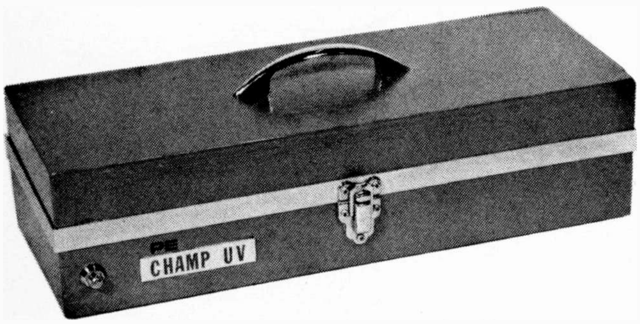
The U.V. lamp bases are fitted to the inside of the lid with the aid of small plywood blocks, and the starter, ballast and a terminal block are secured to the bottom of the case using small woodscrews and nuts and bolts. The microswitch is secured to the inside of the case with small woodscrews, and must of course be carefully positioned to ensure correct operation when the lid is closed. A small plywood fillet is mounted in the case lid to actuate the microswitch.

Wiring up should be carried out with mains quality flexible wire in accordance with Fig. 10.5. Notice that the aluminium panels *must* be connected to the mains earth for safety reasons.

When wiring up is completed, the unit can be connected to the mains and tested for correct operation. A conventional fluorescent tube could be substituted for the U.V. version during the testing phase if it becomes necessary to operate the lamp with the lid open.

The prototype case was finished with aerosol paints and polyurethane varnish using the same techniques as before on CHAMP and CHAMP-PROG. Finally, a carrying handle should be screwed to the lid of the case, and an attache case latch screwed to the front to hold the lid closed for transportation purposes.





FREE OFFER NOTICE

No further Intel Programming manuals are available, but many MCS Users Manuals remain. These will continue to be sent free of charge on receipt of an 8 x 10 inch envelope with 25p stamp (32p 1st class).

USING CHAMP-U.V.

Complete erasure of PROMS prior to reprogramming is absolutely essential for reliable operation. The erasure process is a linear one and does not occur suddenly, and so even when a cell location appears to be erased, further exposure may be required to reach a satisfactory level of gate discharge.

A CHAMP-U.V. system built as described will probably erase all 4702A type PROMS satisfactorily if they are given a 20 minute exposure, but for greater accuracy, system calibration can be an advantage. This is achieved by programming a PROM with all "ones" (FFH in every location) and then giving it short erase increments of say, 2 minutes; checking after each increment for proper erasure. When the chip appears to be completely erased, note the time required and in normal practice always use a cycle of *five times* that duration.

This calibration need only be carried out once, since PROMS are very consistent in their requirements, and a factor allowing for ageing in the lamp tubes has been incorporated.

PROM PREPARATION

Before erasure, always check the quartz window on the PROM for any dirt particles which may cause shadows on the chip, and also wipe them over with a swab soaked in methylated spirit to remove any films which may be opaque to U.V. light. This latter precaution is particularly necessary if gummed labels have been used to cover the PROM when in use.

CHAMP-U.V. is quite capable of erasing any U.V. sensitive PROM including the larger capacity 2704 and 2708 types. The 2704/2708 chips do, however, use a different technology and generally require exposure periods of up to one hour for correct erasure. If CHAMP-U.V. is to be used to erase these devices exclusively, a clockwork timer with a one hour endurance could easily be substituted instead of the thirty minute unit specified earlier.

CONCLUSION

This brings us to the end of the CHAMP series, and we feel sure that everyone who has successfully built all, or part, of the CHAMP system will share our enthusiasm in the results obtained. Our CHAMP is in constant use and has been used to develop several dedicated systems and some "just for fun" software. There are of course many possibilities for additions to the basic system, and we will be glad to hear from anyone who has specific needs which may be catered for in subsequent articles, and from those who wish to pass on programs or hardware designs to their fellow CHAMP enthusiasts.

Meanwhile, the CHAMP programming service for CHOMP, PROMPT, and user programs will continue to be available as long as it is required.

Good luck, and successful programming! ★