PE CHAMP

R.W.COLES B.CULLEN

PART NINE

WITH the circuit details of CHAMP-PROG behind us, this month we can move on to consider the construction of the main board, power supply and plinth, and to an examination of the PROMPT firmware program. Anyone who has already built CHAMP itself should have no problems with CHAMP-PROG, because the techniques required will have already become familiar.

CIRCUIT BOARD

Most of the CHAMP-PROG circuitry is mounted on a piece of Veroboard measuring 165mm by 225mm, and this has to be cut from a larger sheet, of the same type as was used for the CHAMP main board. When the board has been cut to size, the three unperforated copper strips which run along the two long edges should be removed by easing up each of their ends with a sharp knife and then carefully pulling them away from the board along their entire length. The removal of these strips makes for easier insertion of the board into the card guide supports, and insertion can be further eased by chamfering the four board corners with a fine file.

The required track cuts and component positions are detailed in Fig. 9.3 and it is a good idea to study the component layout at this stage so that when construction starts you will "know-your-way-around". There are three distinct circuit areas on the board, the top quarter being occupied by the regulator components, the next quarter by the timing generator and the lower half by the CHAMP interface and the data and address circuits. The layout chosen provides plenty of room in which to work and makes fault finding fairly easy.

Track cuts can be made right at the start, which makes life easier later but requires great care initially to prevent errors creeping in. Alternatively they can be incorporated as construction proceeds so that some layout flexibility is retained.

CONNECTING UP

The circuitry on the CHAMP-PROG board is a fair mixture of digital and analogue integrated circuits and discrete components, and of course high voltages will be present during operation. Needless to say, great care must be taken during wiring-up to avoid expensive mistakes. A wiring error on the prototype caused a transistor to quite literally "blow its top" when power was first applied! One consolation though, faults on this kind of circuitry are usually easy to locate, just watch out for the smoke signals!

As with the CHAMP main board, Soldercon pins are recommended for every integrated circuit, but not for the 16 and 24 way connector socket positions where standard or low-profile sockets are best. Remember to leave the bandolier attached to the Soldercon pins until construction is complete, and be sure not to plug in the 4265 MOS chips until the debugging process is over.

Using the Soldercon pins and sockets as a reference framework, wiring up is carried out using Fig. 8.2. Kynar wire is highly recommended for the interconnection of all logic circuitry address and data drivers, although sturdier single core PVC covered wire is better for the +5V, −10V and +80V interconnections because of its higher current rating and higher voltage insulation.

TESTING

The complex timing generator and voltage regulator circuitry lends itself well to being tested in isolation without benefit of PROMPT software or 4265 interface chip. Before testing can take place, +5V and −10V supplies must be connected, and the 80V supply will have to be built using the circuit of Fig. 9.1 and the layout shown in Fig. 9.2.

To start the testing procedure, first connect pin 1 of IC1 to 0V temporarily to enable the timing generator to free-run (this can be achieved by grounding pin 11 of the vacant IC8 socket if desired). Next apply the +5V and −10V power, but *not* the +80V supply and examine the timing generator waveforms at the Q and $\overline{Q}$ outputs of the 74123 using an oscilloscope set to measure pulse amplitudes of a few volts and pulse durations of a few milliseconds.

At IC1 pin 13 you should be able to see narrow pulses with a 15ms separation. If the pulses you see are separated by much more or less than this, the value of fixed resistor R3 should be changed to compensate. A timing accuracy of ±10 per cent should be the target.

On IC2 pin 13 a series of 3·25ms wide pulses should be obtained, and of course the width of these pulses can be accurately set using VR1. On IC3 pin 12 the pulses should be set to a width of 3·0ms using VR2, and on IC2 pin 12 pulses about 60μs wide should be observed. Finally on IC3 pin 13, pulses of about 155μs width should be visible.

REGULATOR OUTPUTS

If the monostable circuits are operating correctly and VR1 and VR2 have been properly adjusted, the next step is to set VR3 to its mid-travel position (Remember that all three adjustment pots are of the 10 turn variety) and connect up the +80V supply. Providing that the

fuse does not blow (and that no wisps of smoke are observed!), the next step is to examine the 7405 outputs, IC4 pins 2, 4, 8, and 10 and compare these with Fig. 8.5 published last month.

If these drive pulse outputs are correct switch the 'scope probe to the junction of D7 and D8 in the regulator area and decrease the 'scope sensitivity to show pulses of about 50V. The waveform at this point should consist of a steady +4·5V level with pulses 3·25ms wide superimposed every 15·0 ms. The amplitude of these pulses can be set by means of VR3, and this should be adjusted to give a peak of +47V. (Note that the +47V should be measured with respect to 0V and *not* with respect to +4·5V.)

At this point you can relax a little, because the worst is over! All that remains is to check the remaining Cs, Vbb, Vgg, Prgm and Vdd outputs and to ensure that they all conform to the timing and amplitudes specified in Fig. 8.5 last month. The Prgm pulse must have the characteristic "two-eared" shape and you will find that final trimming of this pulse can be achieved using VR1 and VR2. The high voltage programming waveforms can all be found on the board-mounted 24 pin socket as well as on the regulator transistors themselves of course.

CASE CONSTRUCTION

The CHAMP-PROG case, or plinth, is made of plywood and aluminium and is relatively easy to construct using the techniques described in part four for the CHAMP case. The plywood framework should be pinned and glued together, and mated carefully with the aluminium top cover and the separate aluminium back panel. When the overall fit is satisfactory all necessary holes can be drilled in accordance with Fig. 9.4 and the board runners made up and pop riveted or bolted in place on the cover.

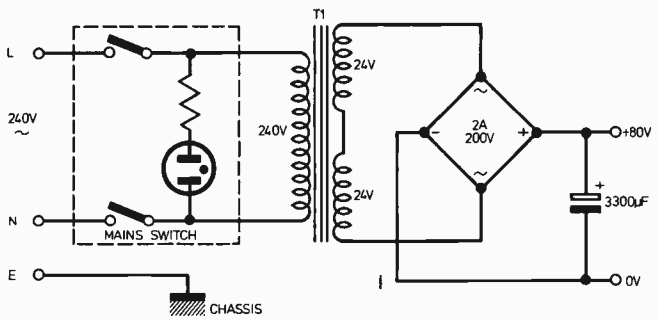


Fig. 9.1. Circuit diagram of 80V supply

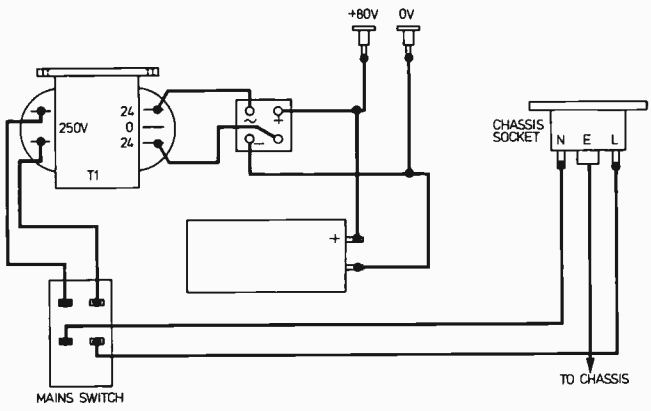


Fig. 9.2. Component layout of 80V Supply

It is of course essential at this stage to ensure that the board runners are positioned so as to provide a satisfactory sliding fit on the CHAMP-PROG circuit board. A great deal of care was taken to ensure that the CHAMP-PROG prototype finish matched that of CHAMP itself, and again the process of applying several coats of primer, sanding it smooth and finishing off with a couple of colour coats, was followed. After allowing the paint to harden for a couple of days Letraset lettering was applied along with outlines drawn with a spirit based pen. Finally a coat or two of polyurethane clear varnish was applied to bring out a high gloss and to protect the lettering.

ZERO INSERTION FORCE SOCKET

With the plinth hardware completed, overall assembly can begin with items such as the ON/OFF rocker switch and the zero insertion force socket mounted on the front panel section of the cover.

If an economy CHAMP-PROG is required, the zero insertion force socket *could* be left out, and PROMS programmed directly in the 24-pin socket on the board, but levering expensive PROMS in and out of this type of socket is less than satisfactory, and the sheer convenience of the lever action type is well worth the few pounds it costs. If used, the front panel socket can be mounted either by adhesive, or more securely by first removing the two small Phillips screws from the socket, and with the socket lever in the upright position removing the face of the socket so that two holes can be drilled at the top and bottom of the socket. These should clear the two Phillips screw holes to take two 8BA countersunk screws. Using the socket as a template, two 8BA clear holes can be drilled in the plinth cover so that the socket can be mounted securely and its faceplate replaced.

EIGHTY VOLT SUPPLY

The programming voltages are derived from a simple power supply which consists of a transformer with a 25-0-25V 2A secondary, a 200V 2A bridge rectifier, and a 3,300μF, 100V electrolytic capacitor. These bulky components are mounted inside the plinth using the separate aluminium back panel as a support and as a heat sink for the transformer and the rectifier. Mains input is via a three pin connector and, of course, the rocker switch on the front panel. The 80 volt output is routed to two wander sockets on the back panel, and the CHAMP-PROG board connects to these via a couple of flying leads.

The 80V generated by this circuit is of course sufficient to give the unwary quite a tingle, and caution is advisable when making the back panel connections! A fully insulated connector could of course be used instead of the Wander plugs and sockets if required. Since the 80V is also present on the CHAMP-PROG circuit board some protection against prying fingers has been provided by mounting a tailored sheet of perspex over the parts of the board where danger exists. Five 4BA plastic mounting pillars were cut to size and cemented to the Veroboard using cyanoacrylate adhesive. The perspex safety cover is screwed to these pillars when construction is complete.

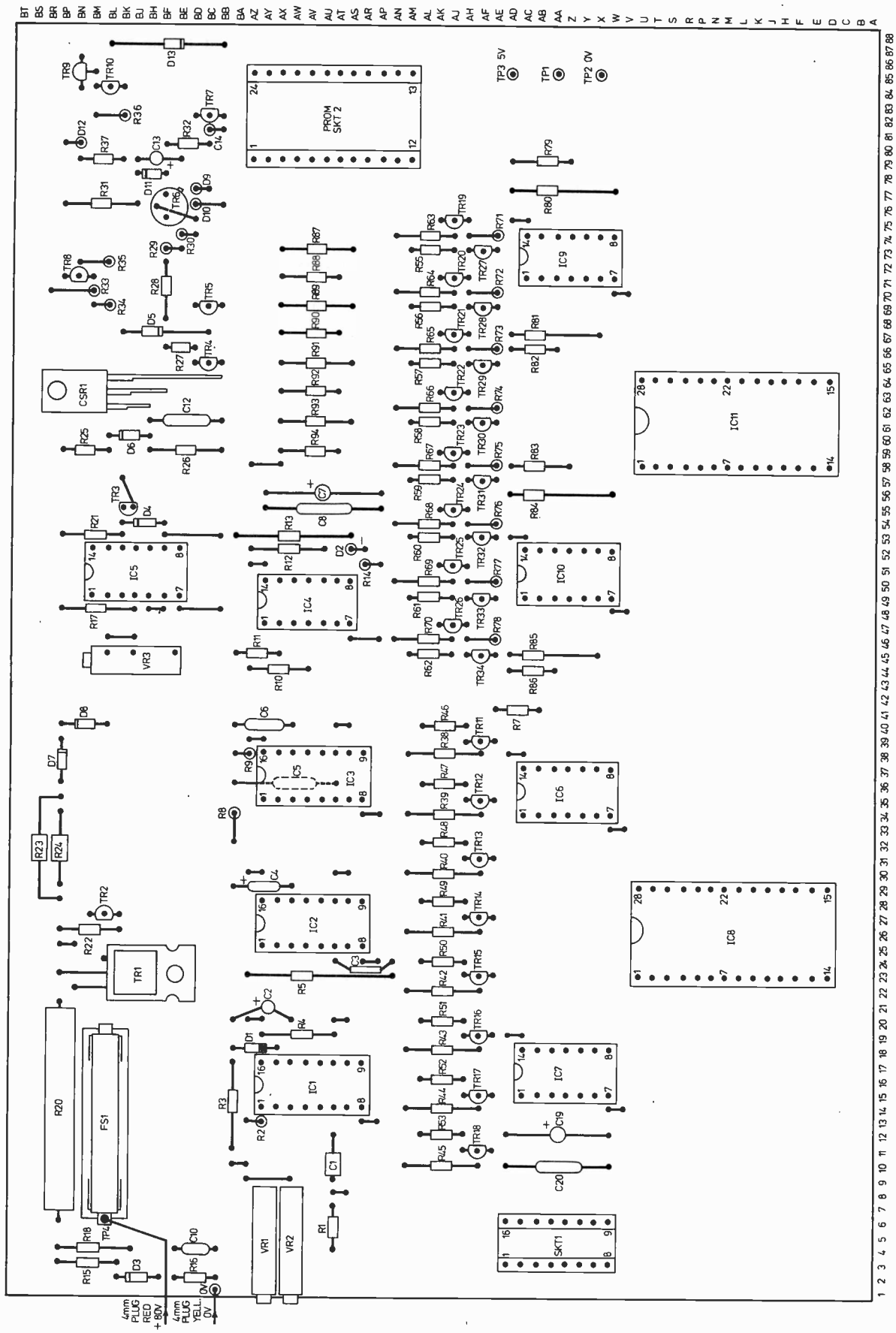


Fig. 9.3. CHAMP-PROG component layout. For interconnections see Fig. 8.2.

Row	Positions
4	X, Y, Z, AA, AB, AC, BK
5	AT, AV, AX
9	X, Y, Z, AA, AB, AC
11	AS, AT, AU, AV, AW, AX
12	BA
13	AF, AH, AL, BP
16	AR–AY
17	W, X, Y, Z, AA, AB, AC, AF
20	BL
21	AF, AH, AL
22	AN, AR–AY, BA
24	BK, BL, BM, BP
25	AF, AH, AL
26	D–Z, AA, AB, AC
27	AR–AY
28	AN, BN
29	AF, AH, AL, BJ, BK
30	BL
31	AR–AY, BA
33	AF, AH, AL, BA, BP
35	W–Z, AA, AB, AC, BE, BF, BH, BJ
37	AF, AH, AL, AR–AY, BA
39	BM
42	AR–AY
43	D–U, W–Z, AA, AB, AC, BA, BE–BN
44	AF–AK
47	BL, BM
48	AF–AK, AN, AR–AY
50	W–Z, AA, AB, AC
51	BE–BJ, BM
52	AF–AK, AN
56	AF–AK, AN, BE, BH–BN
57	AR–AY
60	AF–AK, AN, AT, BC, BD
61	D–U, W–Z, AA, AB, AC
63	AS, BJ
64	AF–AK, AN, AT
65	BN
66	AS, BA
68	AF–AK, AN, AT, BD
70	AS, BE, BF
72	W–Z, AA, AB, AC, AF, AH, AJ, AK, AN, AT, BD
74	BN
75	AN–AZ
76	BF, BK
79	BC, BD, BF
81	BN
82	AM–AZ
84	X, AA

Table of track cut positions on CHAMP-PROG Veroboard

TWENTY-FOUR WAY CONNECTOR

The zero insertion force socket on the front panel is connected via a flying lead to the socket on the circuit board, and in addition to the 24 wires required for the PROM three others are needed for the "PROM POWER" switch and l.e.d. On the prototype a 27-way wiring loom was made up with the three extra wires being terminated at the board end with individual sleeved Soldercon sockets which provided a very convenient means of connection to terminal pins soldered to the Veroboard (Fig. 9.5).

The 24-pin plug required was actually made using a "header plug with top" which is available from Doram. Fine flexible wire was used for the interconnection, and this was soldered to the plug pins and brought out through a hole cut in the right hand side of the header plug top. When fully wired the header plug pins were potted in quick-set Araldite and the top clamped in position until the epoxy hardened. This method of construction has provided a satisfactory and trouble free plug which is a less expensive but more time consuming alternative to the flat-strip cable connectors used on CHAMP itself.

PROMPT FIRMWARE

When construction is complete, and the timing circuits and voltage regulator outputs have been set up correctly, you are ready to plug in the 4265 chips and run the PROMPT firmware program using the control sequence detailed last month. The 4702A containing PROMPT must be in the Chip One socket on the CHAMP main board, and arrangements have been made to enable CHAMP-PROG constructors to get their own devices programmed with PROMPT by using the CHAMP PROM programming service. Of course, once CHAMP-PROG is operating with PROMPT, CHAMP programmers will be totally independent and will never again have to rely on outsiders for programming facilities. The full listing of PROMPT is given in Fig. 9.7 and as you can see there is no wasted space in the 256 line PROM. All users of CHAMP will find it useful to study the operation of the PROMPT software, and to make this easier to follow, some words of explanation might be helpful.

DESIGN AIMS

A prime objective of the software design was that it should make CHAMP-PROG simple to use and preferably self explanatory. To this end the program has been made "interactive" with the programmer. PROMPT issues a prompting message via the keyboard display and waits for a response from the programmer, this process is repeated three times for address entry. At the termination of a programming run, CHAMP-PROG will issue the message "done" or "fail".

For address entry the CHOMP keyboard interrupt routine is "borrowed", and for the display of messages and keyboard entries the CHOMP DDRV subroutine is used. Here is a good example of why it pays to make software routines as general-purpose as possible from the outset, and to code them as subroutines callable from anywhere in program memory! The DDRV subroutine is segment, rather than BCD based and so it is quite capable of refreshing a display of alphanumeric characters when required. The generation of text messages is of course a new facility, and PROMPT includes a new subroutine, TEXT, to handle this job.

MAIN FLOWCHART

Referring to Fig. 9.6 PROMPT is entered at the top via a JUN instruction, from CHOMP. Since CHOMP itself was unaware of the presence of CHAMP-PROG during CHAMP initialisation, the first job here is to set the modes of the two new 4265 chips via WMP instructions (box 1). Next, (box 2) the prompt message "Adr 1" is loaded into the display RAM buffer register by the subroutine TEXT, and then an interruptible display loop is entered to await keyboard response (boxes 3 and 4). The continuous looping automatically refreshes the display via DDRV, while accepting up to three hexadecimal digits via the INTER routine.

An exit from the loop is made via box 4 by pressing the ENTER DATA button. At this time, a three digit hexadecimal address should be resident in 4040 registers C, D and E, and of course visible on the left of the display. The next job is to store Adr 1 away in its appointed storage locations (box 5) and to change the display message to Adr 2 (box 6). In this case it is not necessary to change the message radically, and so rather than employ TEXT once more, the display buffer is modified directly to save program lines. Boxes 7 and 8 are of course identical to boxes 3 and 4 also boxes 11 and 12 further down, and this makes them ideal subroutine candidates. (In fact a subroutine ENTERL *does* contain this pair of boxes, but for the purposes of our flow chart these activities have been included individually, to clarify program action.)

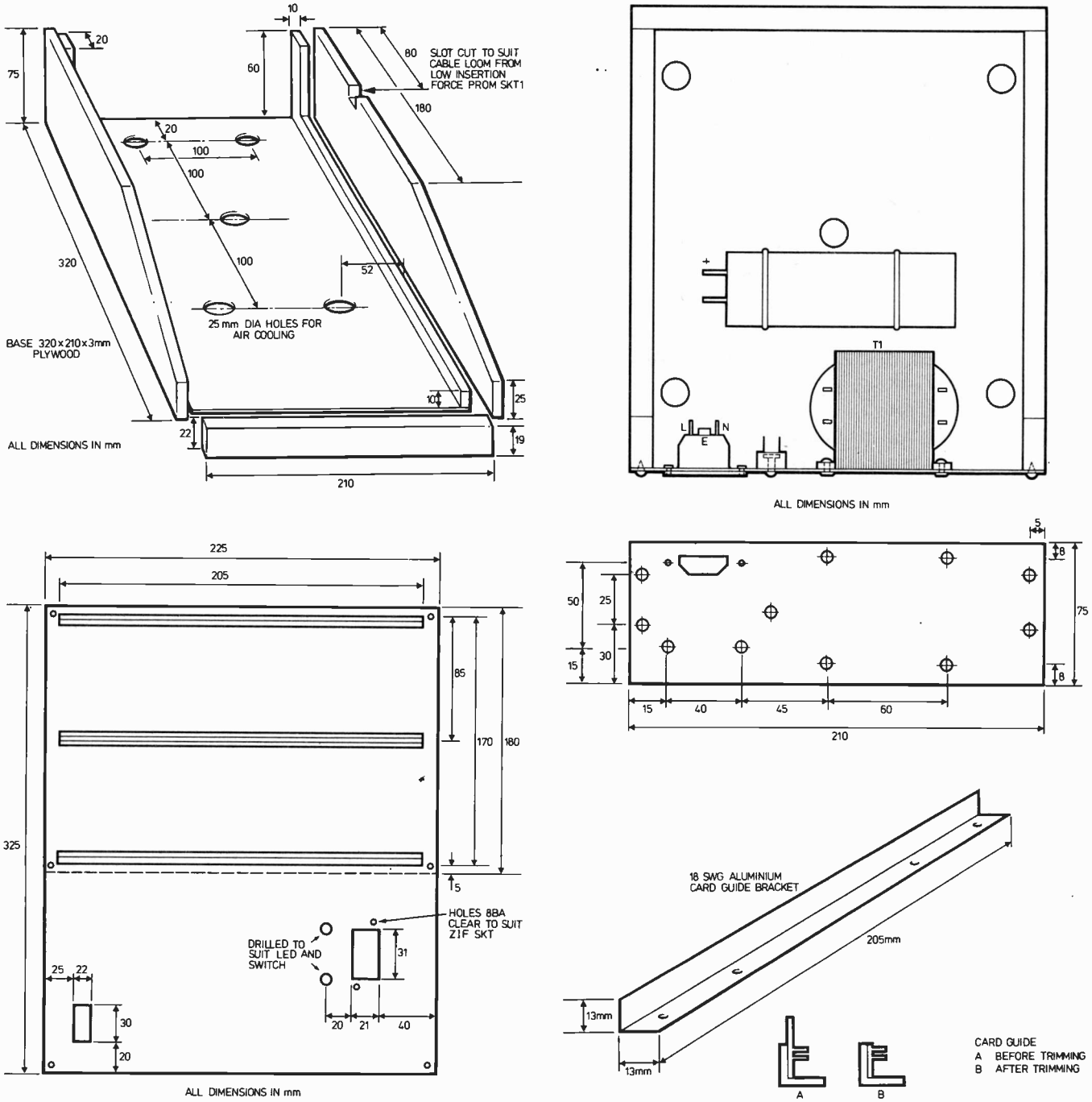


Fig. 9.4. Physical dimensions of CHAMP-PROG chassis

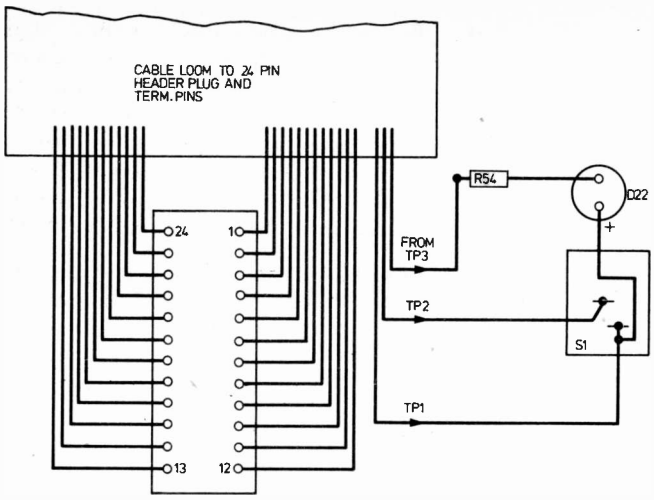


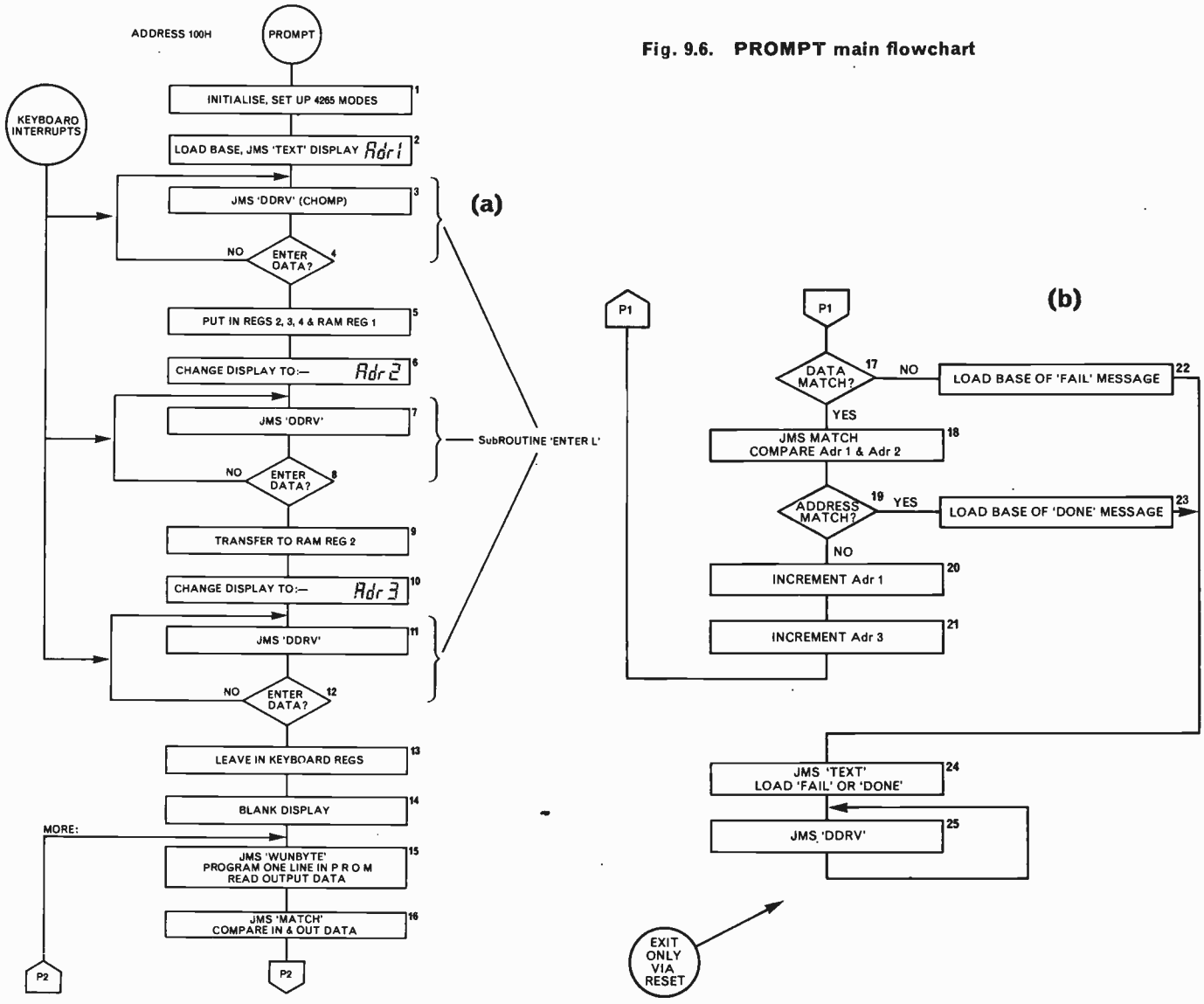
Fig. 9.5. Cable loom arrangement for z.i.f. socket

With three new hexadecimal digits entered, ENTER DATA is pressed once more and the Adr 2 data stored away in RAM register 2 where it will later be needed for comparison with Adr 1 (box 9). The display is now modified to show Adr 3, and the interruptible display loop again used for address entry. Adr 3 data is left in the 4040 registers C, D and E, and so box 13 does not so much represent an *action* but rather it is a reminder that no action takes place in this case.

Next the keyboard display is blanked by writing OOH to ports X and Y of the CHAMP 4265 so that in the absence of a refresh loop, a single display digit does not remain on continuously.

Box 15 represents a largish subroutine called WUNBYTE which has the job of programming a single PROM location each time it is called. This subroutine reads data from the appointed area in program RAM, sends it out together with Adr 3 to the CHAM-PROG 4265 chips, initiates a program cycle via 4265 number 1 port Z3, and waits during a software delay of about 540 milliseconds before reading the results of its programming and storing them away in RAM register 2 before returning to the main program.

Fig. 9.6. PROMPT main flowchart



In box 16 a subroutine called MATCH is used to compare a copy of the PROM input data stored in RAM with the output data read by WUNBYTE after its cycle. MATCH returns a flag which is tested in box 17, where a conditional jump (JCN) either aborts operations if the output is bad, or passes on to an address compare operation using MATCH (box 18) if the output is good.

Address comparison is needed to check whether all necessary locations have been programmed. In this case further programming is suspended if a *good* comparison results. If on the other hand Adr 1 and Adr 2 are not yet the same, then Adr 1 and Adr 3 are incremented to point to the next source address and the next destination (PROM) address respectively, before a JUN loop to the label MORE is carried out (boxes 20 and 21).

Boxes 22 and 23 load the base addresses of text messages stored in PROM before jumping to a routine which loads the message into the display using TEXT

PROMPT

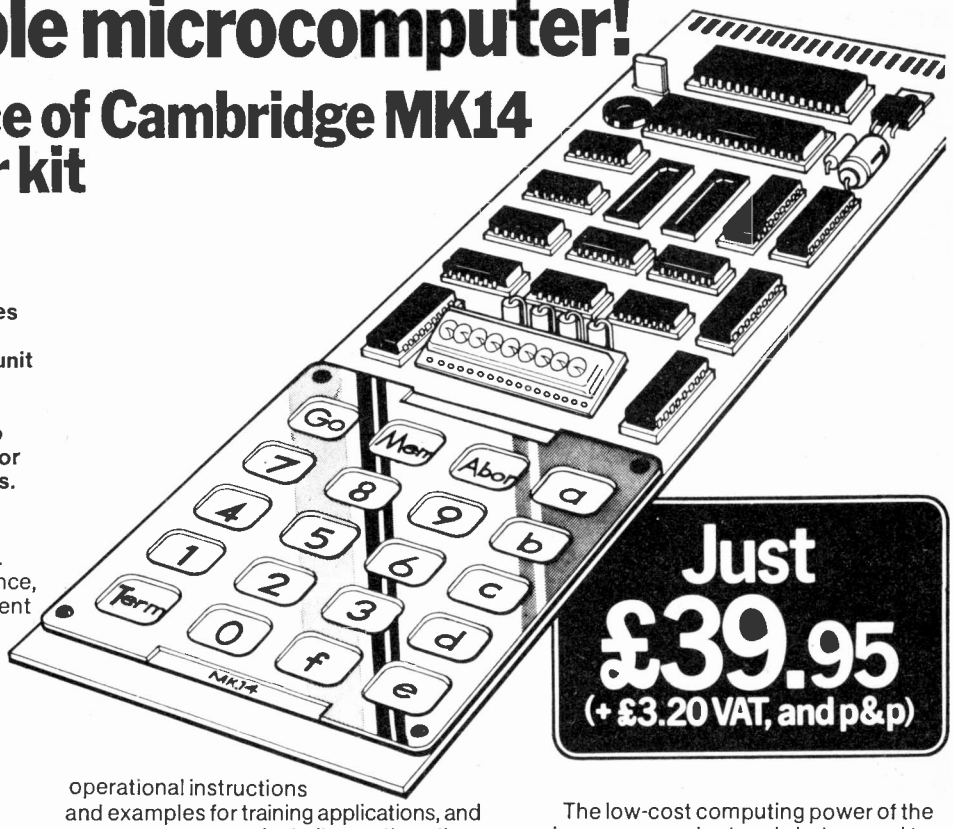
PAGE	LINE	ROM CODING	LABEL	OPERATION	OPERAND	COMMENTS
1	0	D9	PROMPT	LDM	9	LOAD DCL&SRC CODE
	1	FD		DCL		SELECT RAM BANK 1
	2	B8		XCH	8	9H → R8
	3	29		SRC	9	SELECT 4265 CHIP
	4	D4		LDM	4	
	5	E1		WMP		SET TO MODE 4
	6	DF		LDM	F	BIT SET PORT Z3
	7	E0		WRM		
	8	D2		LDM	2	
	9	FD		DCL		SELECT RAM BANK 2
	A	29		SRC	9	SELECT 4265 CHIP
	B	D6		LDM	6	
	C	E1		WMP		SET TO MODE 6
	D	D0		LDM	0	
	E	FD		DCL		SELECT RAM BANK 0
	F	0B		SB1		SELECT REG. BANK 1
1	1	20		FIM	0	SET UP BASE ADDRESS
	2	F4		F	4	OF MESSAGE "Adr1"
	3	51		JMS		LOAD MESSAGE INTO
	4	E1		TEXT		DISPLAY
	5	51		JMS		LOOP FOR Adr1 KEYBOARD
	6	C0		ENTERL		ENTRY & 'ENTER DATA'
	7	AD		LD	D	GET Adr1 LEAST SIG
	8	B3		XCH	3	PUT IN R3
	9	AC		LD	C	GET Adr1 MID
	A	B2		XCH	2	PUT IN R2
	B	AE		LD	E	GET Adr1 MOST SIG
	C	B4		XCH	4	PUT IN R4
	D	51		JMS		COPY LEAST SIG & MID
	E	D0		LADR1		OF Adr1 TO RAM
	F	20		FIM	0	CODE FOR TEXT
		6B		6	B	OF "2"
1	2	51		JMS		CHANGE DISPLAY TO
	1	B6		ADNO		"Adr 2"
	2	51		JMS		LOOP FOR Adr2 KEYBOARD
	3	C0		ENTERL		ENTRY & "ENTER DATA"
	4	28		FIM	8	SET UP Adr2 RAM
	5	22		2	2	ADDRESS
	6	29		SRC	9	
	7	AD		LD	D	LEASTSIG Adr2 TO RAM
	8	E0		WRM		
	9	69		INC	9	
	A	29		SRC	9	
	B	AC		LD	C	MID Adr2 TO RAM
	C	E0		WRM		
	D	20		FIM	0	GET CODE FOR TEXT
	E	E9		E	9	OF "3"
	F	51		JMS		CHANGE DISPLAY TO
1	3	B6		ADNO		"Adr 3"
	1	51		JMS		LOOP FOR Adr3 KEYBOARD
	2	C0		ENTERL		ENTRIES & ENTER DATA
	3	28		FIM	8	
	4	80		8	0	
	5	29		SRC	9	SELECT CHAMP 4265
	6	F0		CLB		
	7	E5		WR1		} BLANK
	8	E6		WR2		} DISPLAY
1	3	51	MORE	JMS		PROGRAM ONE PROM
	A	5A		WUNBYTE		LOCATION
	B	20		FIM	0	SET UP ADDRESS OF
	C	10		1	0	'IN' DATA FOR MATCH
	D	22		FIM	2	SET UP ADDRESS OF
	E	20		2	0	'OUT' DATA FOR MATCH.
	F	51		JMS		COMPARE 'IN' & 'OUT'

PAGE	LINE	ROM CODING	LABEL	OPERATION	OPERAND	COMMENTS
1	4	A6		MATCH		
	1	20		FIM	0	SET UP BASE ADDRESS
	2	F8		F	8	OF TEXT MESSAGE "FAIL"
	3	1C		JNZ		STOP & DISPLAY "FAIL"
	4	DA		FINIS		IF AC NOT ZERO.
	5	20		FIM	0	SET UP ADDRESS OF Adr1
	6	12		1	2	FOR MATCH.
	7	22		FIM	2	SET UP ADDRESS OF Adr2
	8	22		2	2	FOR MATCH.
	9	51		JMS		COMPARE CURRENT Adr1
	A	A6		MATCH		WITH Adr2
	B	20		FIM	0	SET UP BASE ADDRESS
	C	FC		F	C	OF TEXT MESSAGE "donE"
	D	14		JZ		STOP & DISPLAY
	E	DA		FINIS		"donE"
	F	0A		SB0		SELECT REG BANK 0
1	5	73		ISZ	3	}
	1	53		PAST		} INCREMENT Adr1
	2	62		INC	2	}
1	5	51	PAST	JMS		NEW Adr1 TO RAM
	4	D0		LADR1		}
	5	7D		ISZ	D	}
	6	39		MORE		} INCREMENT Adr3
	7	6C		INC	C	}
	8	41		JUN		} LOOP BACK FOR NEXT BYTE
	9	39		MORE		} END OF MAIN PROG.
1	5	28	WUNBYTE	FIM	8	SUBROUTINE PROGS 1 BYTE
	B	00		0	0	
	C	29		SRC	9	SELECT PROGRAM MEMORY
	D	A4		LD	4	SOURCE CHIP.
	E	E1		WMP		
	F	23		SRC	3	SEND OUT Adr1
1	6	0E		RPM		READ FIRST NIBBLE.
	1	F4		CMA		COMPLEMENT IT.
	2	B1		XCH	1	PUT IT IN R1
	3	0E		RPM		READ SECOND NIBBLE.
	4	F4		CMA		COMPLEMENT IT
	5	B0		XCH	0	PUT IT IN R0
	6	28		FIM	8	}
	7	10		1	0	}
	8	29		SRC	9	} PUT 'IN' DATA IN
	9	A1		LD	1	} RAM FOR USE
	A	E0		WRM		} WITH MATCH
	B	69		INC	9	} LATER.
	C	29		SRC	9	}
	D	A0		LD	0	}
	E	E0		WRM		}
	F	D2		LDM	2	
1	7	FD		DCL		SELECT RAM BANK 2
	1	28		FIM	8	}
	2	80		8	0	} SELECT 4265 No 2
	3	29		SRC	9	}
	4	A1		LD	1	}
	5	E6		WR	2	} 'IN' DATA TO PORTS
	6	A0		LD	0	} Y & Z.
	7	E7		WR	3	}
	8	D1		LDM	1	}
	9	FD		DCL		} LOAD Adr3 TO PORTS
	A	AD		LD	D	} W & X 4265 No 1
	B	E4		WR0		}
	C	AC		LD	C	}
	D	E5		WR1		}
	E	DE		LDM	E	TURN ON PROGRAM
	F	E0		WRM		PULSES.

Fig. 9.7. PROMPT program listing

MK14-the only low-cost keyboard -addressable microcomputer!

The new Science of Cambridge MK14 Microcomputer kit



Just £39.95 (+ £3.20 VAT, and p&p)

The MK14 National Semiconductor Scamp based Microcomputer Kit gives you the power and performance of a professional keyboard-addressable unit – for less than half the normal price. It has a specification that makes it perfect for the engineer who needs to keep up to date with digital systems, or for use in school science departments. It's ideal for hobbyists and amateur electronics enthusiasts, too.

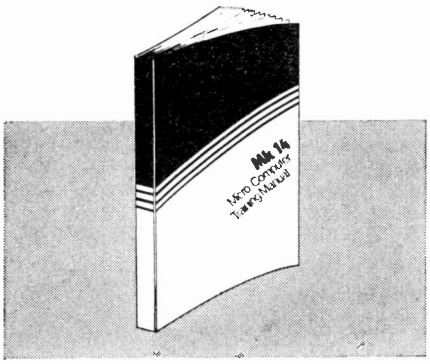
But the MK14 isn't just a training aid. It's been designed for practical performance, so you can use it as a working component of, even the heart of, larger electronic systems and equipment.

MK14 Specification

* Hexadecimal keyboard
* 8-digit, 7-segment LED display
* 512 x 8 Prom, containing monitor program and interface instructions
* 256 bytes of RAM
* 4MHz crystal
* 5V stabiliser
* Single 6V power supply
* Space available for extra 256 byte RAM and 16 port 1/0
* Edge connector access to all data lines and 1/0 ports

Free Manual

Every MK14 Microcomputer kit includes a free Training Manual..It contains



operational instructions and examples for training applications, and numerous programs including math routines (square root, etc) digital alarm clock, single-step, music box, mastermind and moon landing games, self-replication, general purpose sequencing, etc.

Designed for fast, easy assembly

Each 31-piece kit includes everything you need to make a full-scale working microprocessor, from 14 chips, a 4-part keyboard, display interface components, to PCB, switch and fixings. Further software packages, including serial interface to TTY and cassette, are available, and are regularly supplemented.

The MK14 can be assembled by anyone with a fine-tip soldering iron and a few hours' spare time, using the illustrated step-by-step instructions provided.

Tomorrow's technology – today!

"It is not unreasonable to assume that within the next five years ... there will be hardly any companies engaged in electronics that are not using micro-processors in one area or another."

Phil Pittman, Wireless World, Nov. 1977.

The low-cost computing power of the microprocessor is already being used to replace other forms of digital, analogue, electro-mechanical, even purely mechanical forms of control systems.

The Science of Cambridge MK14 Standard Microcomputer Kit allows you to learn more about this exciting and rapidly advancing area of technology. It allows you to use your own microcomputer in practical applications of your own design. And it allows you to do it at a fraction of the price you'd have to pay elsewhere.

Getting your MK14 Kit is easy. Just fill in the coupon below, and post it to us today, with a cheque or PO made payable to Science of Cambridge. And, of course, it comes to you with a comprehensive guarantee. If for any reason, you're not completely satisfied with your MK14, return it to us within 14 days for a full cash refund.

Science of Cambridge Ltd,
6 Kings Parade,
Cambridge,
Cambs., CB2 1SN.
Telephone: Cambridge (0223) 311488

To: Science of Cambridge Ltd,
6 Kings Parade, Cambridge,
Cambs., CB2 1SN.

Please send me an MK14 Standard Microcomputer Kit. I enclose cheque/ Money order/PO for £43.55 (£39.95 + 8% VAT and 40p p&p).

Name _____

Address (please print) _____

_____

_____

_____

PE/5

Allow 21 days for delivery.

Science of Cambridge

CHROMASONIC electronics

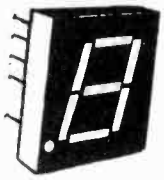
56A Fortis Green Road
Muswell Hill London N10 3HN
Telephone 01-883 3705

C-MOS

4000	·18*	4049	·58*	7442	·52*
4001	·18*	4050	·58*	7445	·70*
4002	·18*	4051	·86*	7447	·59*
4006	1·09*	4052	·86*	7448	·65*
4007	·19*	4060	1·16*	7451	·16*
4009	·62*	4068	·22*	7470	·27*
4010	·62*	4070	·22*	7472	·22*
4011	·18*	4071	·22*	7473	·27*
4012	·21*	4081	·22*	7474	·27*
4013	·53*	4082	·22*	7475	·40*
4014	1·00*	4510	1·24*	7476	·30*
4015	·97*	4511	1·54*	7485	·76*
4016	·58*			7486	·27*
4017	·97*	**TTL**		7489	2·32*
4018	·97*			7490	·38*
4019	·58*			7492	·41*
4020	1·08*	7400	·16*	7493	·41*
4021	·97*	7401	·16*	74107	·27*
4022	·88*	7402	·16*	74121	·27*
4023	·19*	7403	·16*	74123	·49*
4024	·78*	7404	·17*	74141	·65*
4025	·19*	7408	·19*	74145	·65*
4026	1·76*	7410	·16*	74151	·49*
4027	·58*	7413	·27*	74154	·85*
4028	·87*	7420	·16*	74174	·72*
4029	1·18*	7427	·28*	74192	·86*
4040	1·08*	7430	·16*	74193	·86*
4042	·83*	7441	·56*	74196	·97*

You can combine C-MOS & TTL for quantity discounts.
25 plus less 5%. 100 plus less 10%.

SEVEN SEGMENT DISPLAYS



LIGHT EMITTING DIODES

	1+	10+	100+
0·125" (inc. clip)			
TIL209 Red	15p*	13p*	11p*
TIL212 Yellow (H.B.)	27p*	22p*	17p*
TIL216 Red (H.B.)	27p*	22p*	17p*
TIL232 Green (H.B.)	27p*	22p*	17p*
0·2" (inc. Clip)			
TIL 220 Red	15p*	13p*	11p*
TIL224 Yellow (H.B.)	30p*	26p*	20p*
TIL228 Red (H.B.)	30p*	26p*	20p*
TIL234 Green (H.B.)	30p*	26p*	20p*
(H.B.)—High Brightness.			

LITRONIX All Red

DL701	Com.A 0·30"+1	1·72*	DL707	Com.A 0·30" L/H dec. pt.	1·72*
DL704	Com.C 0·30" R/H dec. pt.	1·72*	DL707R	Com.A 0·30" R/H dec. pt.	1·72*
DL721	Com.A 0·51" R/H Dou. dt	3·04*	DL727	Com.A 0·51" R/H Dou. dt.	3·04*
DL746	Com.A ·630"+1	2·43*	DL750	Com.C ·630" L/H dec. pt.	2·43*
DL747	Com.A ·630" L/H dec. pt.	2·43*			

MONSANTO Green

MAN51	Com.A 0·30" R/H dec. pt.	2·34*	MAN81	Com.A 0·30" R/H dec. pt.	2·34*
MAN52	Com.A 0·30" L/H dec. pt.	2·34*	MAN82	Com.A 0·30" L/H dec. pt.	2·34*
MAN53	Com.A 0·30"+1	2·34*	MAN83	Com.A 0·30"+1	2·34*
MAN54	Com.C 0·30" R/H dec. pt.	2·34*	MAN84	Com.C 0·30" R/H dec. pt.	2·34*

Red — **Orange**

MAN71	Com.A 0·30" R/H dec. pt.	1·52*	MAN3610	Com.A 0·30" R/H dec. pt.	2·17*
MAN72	Com.A 0·30" L/H dec. pt.	1·52*	MAN3620	Com.A 0·30" L/H dec. pt.	2·17*
MAN73	Com.A 0·30"+1	1·52*	MAN3630	Com.A 0·30"+1	2·17*
MAN74	Com.C 0·30" R/H dec. pt.	1·52*	MAN3640	Com.C 0·30" R/H dec. pt.	2·17*

XCITON

XAN362	C.A. R 0·30" L/H dec. pt.	1·81*	XAN662	C.A. R 0·60" L/H dec. pt.	2·03*
XAN352	C.A. G 0·30" L/H dec. pt.	2·21*	XAN652	C.A. G 0·60" L/H dec. pt.	2·43*
XAN82	C.A. Y 0·30" L/H dec. pt.	2·43*	XAN682	C.A. Y 0·60" L/H dec. pt.	2·03*

SUPER SAVER

★ 555 Timer ★
4 for £1·00*
★ 28p* each ★

OUR NEW A4 I.C. BOOKLET Supplied FREE with orders of any I.C.'s worth £4·00 or more. Contains Circuits, pin connections & Data. (35p inc. P&P if sold alone).

LINEAR I.C.'s AUDIO I.C.'s

MC1352P/ TDA1352	1·20	SN7653N2	1·55
MC1327P/ TDA1327	1·50	SN76552N	·65
MC1330P	1·15	SN76660N	·65
MC1350P	1·00	SN76666N	1·00
MC1351P	1·00	TAA263	2·50
MC1357P/ LM2111N	1·70	TAA320A	·75
MC1375P	1·50	TAA350A	2·70
MC1456CG	2·50	TAA570	2·50
MC1495L	4·90	TBA120S	1·00
MC3302P	1·65	TBA231	1·00
MFC6030A	1·00	TBA520Q	2·00
NE540L	2·25	TBA530Q	2·00
NE546	2·15	TBA540Q	2·00
NE555	·28*	TBA550Q	2·00
NE556	·90*	TBA651	2·25
NE560	4·40*	TBA750Q	2·00
NE561	4·40*	TBA800	1·00
NE562	4·40*	TBA810S	1·10
NE565	1·40*	TBA820	1·10
NE566	1·65*	TBA990Q	2·00
NE567	1·85*	TCA730	4·69
SL414A	1·90	TCA740	4·69
SL415A	2·47	TCA940	1·75
SL440A	2·38*	TDA2020	3·50
SN75491N	·95*	UAA170	2·75*
SN75492N	1·15*	UAA180	2·75*
SN76001N	1·40	ZN414	1·10
SN76003N	2·45	ZN417E	1·68
SN76013N	1·60	ZN423T	1·08*
SN76023N	1·60	ZN424E	1·35
SN76033N	2·45	ZN425E	3·78
SN76544N	1·55	ZN1034E	2·03
		ZN1040E	8·43
		ZNA116E	6·75*

I.C. SOCKETS

8 pin	10p*	16 pin	14p*	28 pin	30p*
14 pin	12p*	24 pin	26p*	40 pin	44p*

I.C. pins 100 54p* 1,000 £3·78*

LINEAR I.C.'s AUDIO I.C.'s

uA703 (TO99)	1.00	uA711 (14 pin DIL)	·90
UA709 (8 pin DIL)	·41	uA711 (TO99)	·98
uA709 (14 pin DIL)	·41	uA741 (8 pin DIL)	·25
uA709 (TO99)	·50	uA741 (14 pin DIL)	·42
uA709 (8 pin DIL)	·50	uA741 (TO99)	·45
uA710 (14 pin DIL)	·55	uA847 (14 pin DIL)	·90
uA710 (TO99)	·60	uA748 (8 pin DIL)	·42

LINEAR I.C.'s AUDIO I.C.'s LINEAR I.C.'s AUDIO I.C.'s

uA 748 (14 pin DIL)	·60	CA3089/TDA1200	2·30	LM372N	1·75		
uA748 (TO99)	·65	CA3090Q	4·25	LM373N	2·99		
uA753 (8 pin DIL)	1·95	CA3123E/LM1820	1·50	LM377N	1·95		
AY-1-0212	6·50	CA3097E	1·75	LM380N-8	1·00		
AY-1-5051	1·65	CA3130S	·99	LM380N	·99		
AY-5-1224A	3·39*	CA3140S	·99	LM381N	1·75		
AY-5-3507	5·99*	CA3160S	1·60	LM382N	1·35		
AY-5-4007	7·99*	CA3600E	1·75	LM3900N	·65		
CA3045	1·50	LM301AN (8 pin)	·40	LM3905N	1·50*		
CA3046	·80	LM301AT (TO99)	·50	LM3909N	·70*		
CA3053	·85	LM308N (8 pin)	·90	MC1303/LM1303N	1·10		
CA3075	2·00	LM308T (8 pin)	1·40	MC1310P	1·80		
CA5080	·80	LM339N	1·00	MC1312P	2·15		
CA3081	1·50	LM370N	2·75	MC1314P	3·50		
CA3082	1·50	LM371	1·75	MC1315P	6·10		

VAT INCLUSIVE PRICES

Postage and Packing 25p
Items followed by a * include V.A.T. @ 8% all others include 12·5% ALWAYS PLEASED TO SEE PERSONAL CALLERS
TRADE AND EXPORT CUSTOMERS MOST WELCOME Overseas Customers deduct 2/27 from items marked with a * 1/9 from others

---

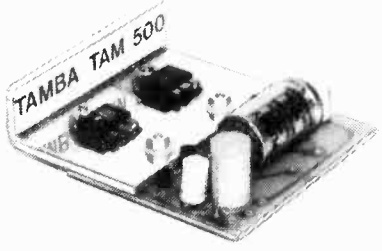
BEFORE *YOU* BUY AN AMPLIFIER MODULE—CHECK:

DOES IT HAVE

★ 30A power transistors ★ 2-year guarantee
★ 3A drivers (100W unit) ★ Integral output capacitor

Then compare with the Tamba range—excellent value—25, 50 and 100W R.M.S

TAM1000 100W 4 ohms 65V	£9·80
TAM500 50W 4 ohms 45V	£7·50
TAM250 25W 8 ohms 45V	£5·75
POWER SUPPLIES	
For 1 or 2 TAM250/500	£7·50
For 1 or 2 TAM1000	£9·80
(Carriage 50p on supplies)	



- Suits loads 4–16 ohms
- 20–20,000 Hz ±1dB
- Silicon circuitry throughout
- Glass fibre P.C.B.
- High sensitivity (100mV 10k)

High grade components used throughout: Texas, Mullard, R.C.A., Plessey, etc.

- Low distortion (0·1%)
- Low profile (1in high 3½in × 3in)
- 75% efficient
- Accepts most mixer/pre-amplifiers
- Four simple connections

NEW

ALL PURPOSE MIXER/PRE-AMP.

(with 60mm slider volume)

- Suitable for multiple input systems
- High and low impedance inputs
- High sensitivity
- Built-in supply smoothing
- 20–20,000 Hz ±1dB
- −80dB noise level
- Accepts a wide variety of inputs
- Wide range bass and treble controls

Use up to 10 PRE-AMPS with 1 power supply

Printed circuit board assembly with treble and bass controls plus slider volume control **£6·50**

You may order as follows: C.W.O. (crossed cheques, P.O.s, M.O.s etc)—C.O.D. (60p extra). We accept Access and Barclaycard—send or telephone your number—do not send your card. Add VAT at 8% to orders for 50 and 100W systems and at 12½% for 25W systems (including preamp if ordered at same time).

Hours, 9.30 a.m.–5 p.m. Monday — Saturday. Callers welcome. Tel. (01) 684 0098

TAMBA ELECTRONICS

Bensham Manor Road Passage, Bensham Manor Road, Thornton Heath, Surrey.

(box 24) and then enters a display loop using DDRV (box 25). From the foregoing you can see that the program always terminates with one of the two messages "fail" or "done" displayed, and that return to CHOMP must be carried out by use of the RESET button.

SUBROUTINE FLOWCHARTS

The flowchart of the ENTERL or "interruptible loop" subroutine is shown in Fig. 9.8, note the use of EIN and DIN to control interrupts, and the way that the ENTER DATA flag is read using RDR and tested using JCN; the WRR before the BBL is used to reset the ENTER DATA flag. Fig. 9.9 shows the TEXT subroutine and its look-up tables. This subroutine is passed the base address of a text message table in register pair O,1 so that the FIN command can be used to fetch the message a byte at a time.

In PROMPT only four-character messages are per-

PROMPT

PAGE	LINE	ROM CODING	LABEL	OPERATION	OPERAND	COMMENTS
1	8	0B		SB1		
		22		FIM	2	
		19		1	9	PRESET DELAY
		24		FIM	4	COUNTER.
		9A		9	A	
1	8	72	WAIT	ISZ	2	
		85		WAIT		
		73		ISZ	3	
		85		WAIT		540M.S. DELAY
		74		ISZ	4	
		85		WAIT		
		75		ISZ	5	
		85		WAIT		
		DF		LDM	F	TURN OFF
		E0		WRM		PROGRAM PULSES.
1	8	74	DELAY	ISZ	4	
		8F		DELAY		DELAY BEFORE
		75		ISZ	5	READ 1.6 M.S.
		8F		DELAY		
		D2		LDM	2	
		FD		DCL		SELECT 4265 No 2
		29		SRC	9	
		EC		RDO		
		B5		XCH	5	READ FROM 'OUT' DATA
		ED		RD1		PUT IN R5, R4.
		B4		XCH	4	
		D0		LDM	0	
		FD		DCL		SELECT RAM BANK 0
		28		FIM	8	CHIP 0 REG 2
		20		2	0	
		29		SRC	9	
		A5		LD	5	
1	A	E0		WRM		
		69		INC	9	PUT 'OUT' DATA INTO
		29		SRC	9	RAM.
		A4		LD	4	
		E0		WRM		
		C0		BBL	0	END OF WUN BYTE.
1	A	FA	MATCH	STC		SUBROUTINE COMPARES 2 BYTES.
		DE		LDM	E	PRESET NIBBLE COUNTER.
		B5		XCH	5	
1	A	F3	LOOP2	CMC		
		21		SRC	1	
		E9		RDM		READ FIRST NIBBLE.
		23		SRC	3	
		E8		SBM		SUBTRACT SECOND NIBBLE.
		61		INC	1	INCREMENT RAM
		63		INC	3	ADDRESSES.
1	B	14		JZ		
		B3		SKIP		
		C1		BBL	1	IF AC. NOT 0 BBL 1
1	B	75	SKIP	ISZ	5	NEXT PAIR OF
		A9		LOOP	2	NIBBLES ?
		C0		BBL	0	IF AC STILL 0 BBL 0.
1	B	28	ADNO	FIM	8	SUBROUTINE MODIFIES DISPLAY
		0E		0	E	SELECT RAM BANK 0,
		29		SRC	9	CHIP 0, CHAR E.
		A0		LD	0	PUT FIRST NIBBLE IN
		E0		WRM		CHAR E.
		69		INC	9	NEXT CHAR
		29		SRC	9	
		A1		LD	1	PUT SECOND NIBBLE
		E0		WRM		IN CHAR F.
		C0		BBL	0	END OF ADNO

PAGE	LINE	ROM CODING	LABEL	OPERATION	OPERAND	COMMENTS
1	C	0C	ENTERL	EIN		SUBROUTINE: KEY ENTRY LOOP
		28		FIM	8	
		40		4	0	READ SWITCH FLAGS.
		29		SRC	9	
		EA		RDR		
		F6		RAR		PUT 'ENTER DATA' IN CY
		0D		DIN		DISABLE INTERRUPTS.
		50		JMS		DISPLAY NEXT DIGIT.
		B1		DDRV		
		12		JC		IF 'ENTER DATA' NOT
		C0		ENTERL		PRESSED THEN LOOP.
		28		FIM	8	NOW PRESSED SO
		50		5	0	CLEAR SWITCH FLAGS.
		29		SRC	9	
		E2		WRR		
		C0		BBL	0	END OF ENTERL
1	D	28	LADR1	FIM	8	SUBROUTINE: ADR1 TO RAM.
		12		1	2	SELECT RAM BANK 0
		29		SRC	9	CHIP 0, REG 1, CHAR 2.
		A3		LD	3	
		E0		WRM		LEAST SIG TO RAM.
		69		INC	9	
		29		SRC	9	NEXT CHAR.
		A2		LD	2	
		E0		WRM		MID TO RAM.
		C0		BBL	0	END OF LADR1.
1	D	51	FINIS	JMS		SUBROUTINE: END OF PROG.
		E1		TEXT		LOAD TEXT TO DISPLAY
1	D	0A	LOOPX	SB0		
		50		JMS		DISPLAY NEXT DIGIT.
		B1		DDRV		
		41		JUN		LOOP UNTIL RESET.
1	E			LOOPX		END OF FINIS.
1	E		TEXT	FIM	2	SUBROUTINE: LOADS TEXT.
				0	8	SELECT RAM BANK 0, CO, R0, C8
				LDM	C	PRESET BYTE COUNT.
				XCH	5	
1	E		LOOP1	FIN	6	LOOK UP IN TEXT TABLE.
				INC	1	NEXT TABLE LINE
				SRC	3	RAM SRC
				LD	7	FIRST NIBBLE TO RAM
				WRM		DISPLAY BUFFER.
				INC	3	NEXT RAM CHAR.
				SRC	3	
				LD	6	SECOND NIBBLE TO
				WRM		RAM DISPLAY BUFFER.
				INC	3	NEXT RAM CHAR.
				ISZ	5	
1	F			LOOP	1	NEXT BYTE ?
				SB0		
				BBL	0	END OF TEXT.
1	F			NOP		
			ADR1	E	E	TEXT TABLE: H
				B	C	d
				A	0	r
				0	C	l
1	F		FAIL	E	2	F
				E	E	A
				6	0	i
				7	0	L
1	F		DONE	B	C	d
				B	8	o
				A	8	n
				F	2	E

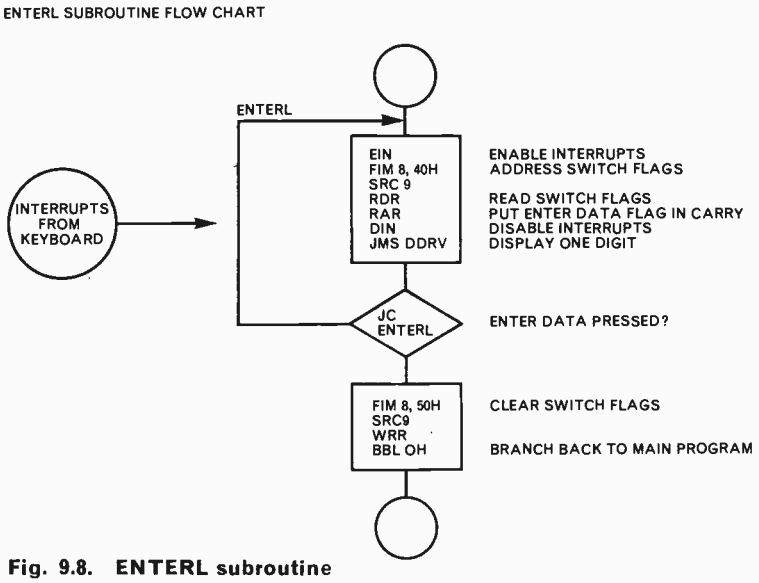


Fig. 9.8. ENTERL subroutine

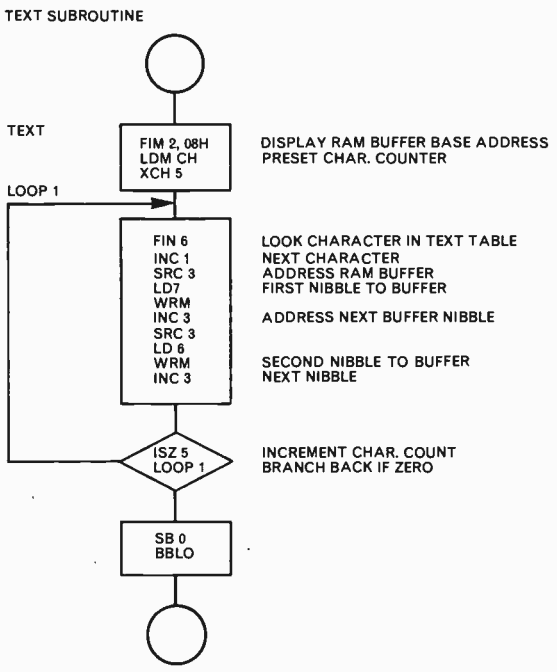


Fig. 9.9. TEXT subroutine

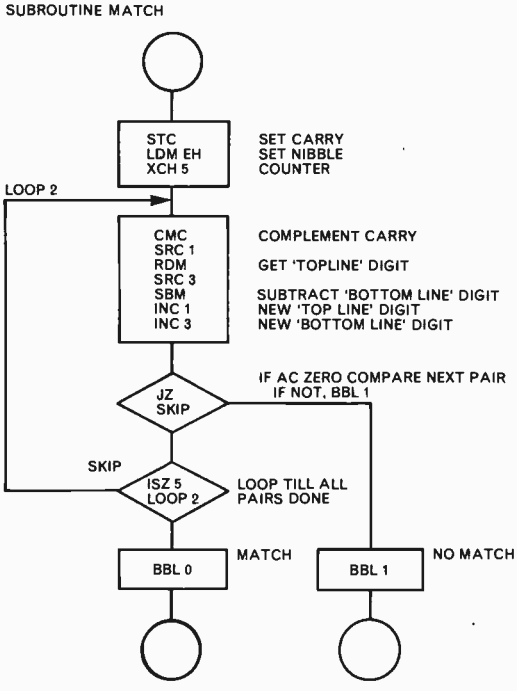


Fig. 9.10. MATCH subroutine

mitted to save program space, but any message of up to eight characters could be used in other applications, with appropriate table entries and a different preset for the byte counter, register 5. The messages themselves were worked out to give the most pleasing display within the restrictions of a seven segment format.

The MATCH subroutine is shown in Fig. 9.10, and here the ploy is to subtract one byte of data from the other and check if the result is zero. The result of the comparison is flagged to the main program via alternative BBL exits, BBL 0 means match, BBL 1 means no match. The addresses of the data to be compared are passed to MATCH in register pairs 0,1 and 2, 3 by the main program.

PROMPT FIRMWARE

Send PROM with remittance of £5·35 to:

C.C. Consultants, Dept. P.E., 3 Gainsborough Drive, Worle, Weston-S-Mare, Avon.

Please ensure PROM is securely packaged, and state clearly whether CHOMP or PROMPT firmware is required.

There is not sufficient room to fully detail WUNBYTE, but since this subroutine is so important, readers may like to draw up their own flowchart using the listing in Fig. 9.7. Fortunately WUNBYTE *is* relatively straightforward and should pose few problems.

Note that source data is complemented using CMA before programming to compensate for the 74L00 inversion. Notice also the way the registers 2, 3, 4, 5 are preset using FIM instructions to give a 16 bit counter which produces a delay of 540 milliseconds. For further details see page 2.17 of the MCS40 manual.

PROMPT LISTING

For PROMPT we have chosen to list the program code in the format introduced in Part 7 for the TONE program. This format differs from the cross assembler listing given in Part 6, and is used to demonstrate to all budding CHAMP programmers that hand-coding of long programs is perfectly feasible! No facilities other than CHAMP itself were used in the development of the PROMPT firmware.

NEXT MONTH: Using CHAMP-PROG and construction of CHAMP-U.V. (Conclusion of series).