Multi-function Intelligent Data Logger

User and Technical Manual

Stock No 611-975

USER AND TECHNICAL MANUAL

MULTI-FUNCTION INTELLIGENT DATA LOGGER

Stock Number 611–975

SECTION 1: GENERAL INSTRUCTIONS

SECTION 2: STANDARD PROGRAM INSTRUCTIONS

SECTION 3: EXTENDED FACILITIES ROM INSTRUCTIONS

SECTION 4: TECHNICAL DETAILS

GENERAL DESCRIPTION

The Logger is microprocessor based and is capable of performing the function of many different items of conventional data logging equipment. All the programs, that are likely to be required are stored in a ROM (read only memory) and can be called by the user by typing in a two digit number using the keypad on the front of the instrument. It is however, possible for the user to write a program for the machine if the stored programs are inadequate for the user's particular requirement.

It is possible to monitor voltages on four input channels. The voltages can be in the range +/–250 mV to +/–25 V. Pulses can be monitored on a separate pulse input channel.

The measurements made can be displayed on an oscilloscope, a chart recorder or on the integral 8-digit display according to the wish of the user and the particular program being used. Data can also be transferred to a microcomputer so that, for example, calculations can be performed on the data.

A diagram of the controls and connectors on the outside of the instrument, and a schematic block diagram of the circuit inside the instrument, appear on the following two pages. The description which follows should be read in conjunction with these diagrams.

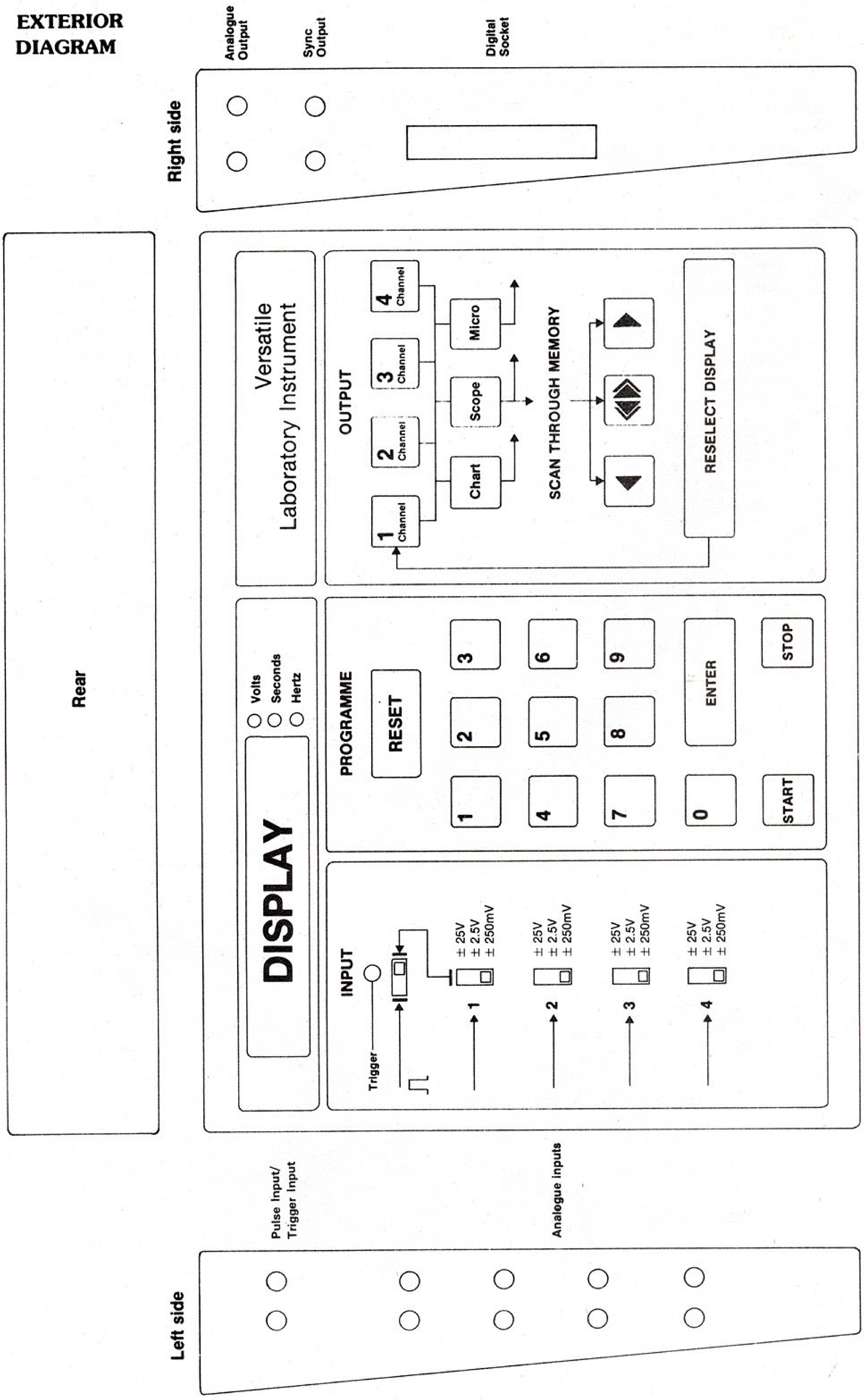
1 ANALOGUE INPUTS

Data which comes in the form of a variable voltage (eg from temperature sensors, pH meters, measurements in electric/electronic circuits) is fed into the instrument via these inputs, which are on the left hand side of the instrument. The 4 mm sockets will take BNC adapters, RS stock number 456-009.

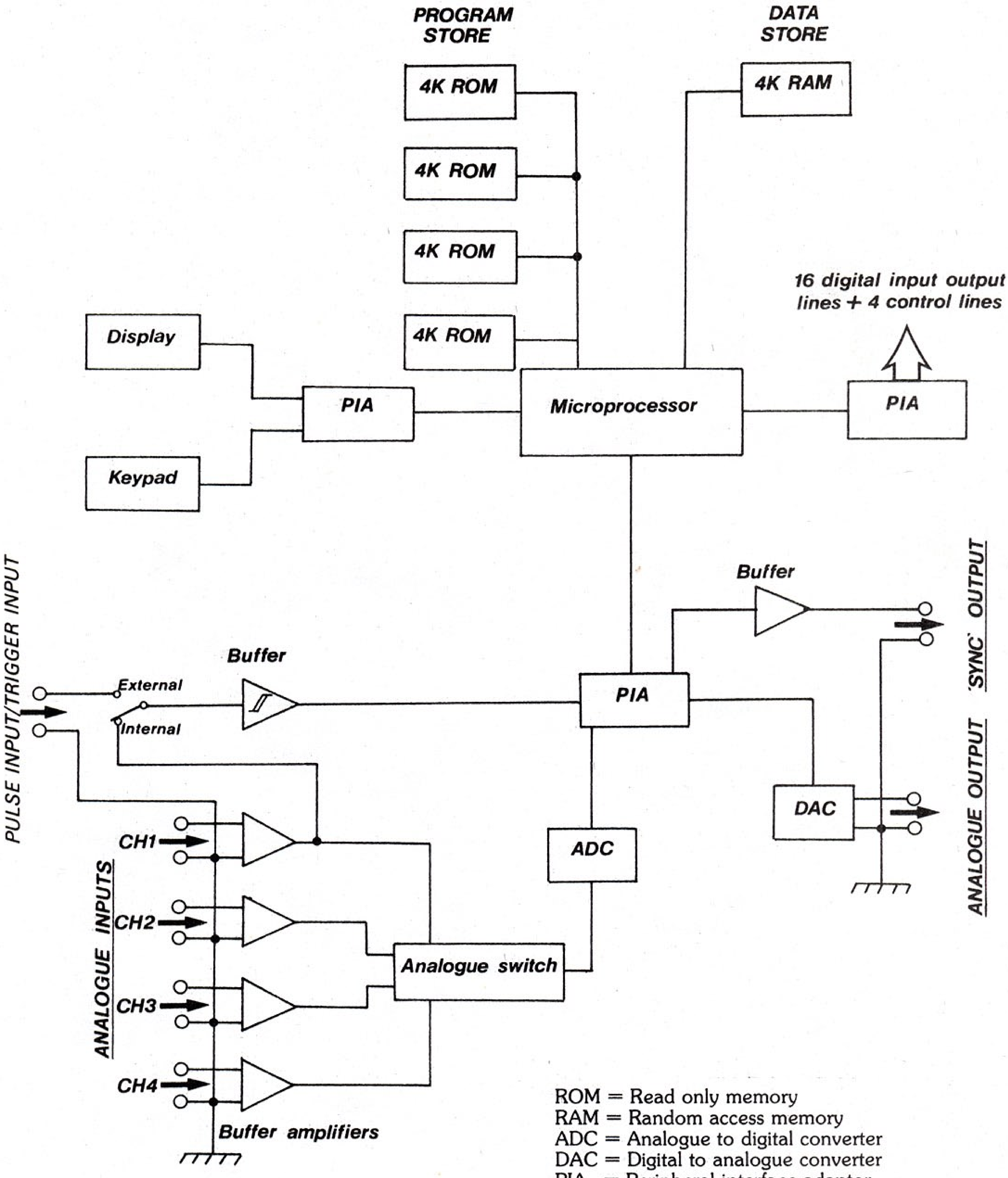
It can be seen from the diagram that each of the four analogue inputs is connected to an amplifer. This amplifier provides three input ranges: +/–25 V, +/–2.5 V and +/–250 mV. The desired range is selected by a switch, which is on the left hand side of the front panel.

The input of each analogue channel has an input impedance of 1 M ohm and is protected up to a maximum voltage of 250 V, but voltages in excess of +/– 25 V should not be exceeded.

EXTERIOR DIAGRAM



BLOCK DIAGRAM OF THE INSIDE OF THE INSTRUMENT



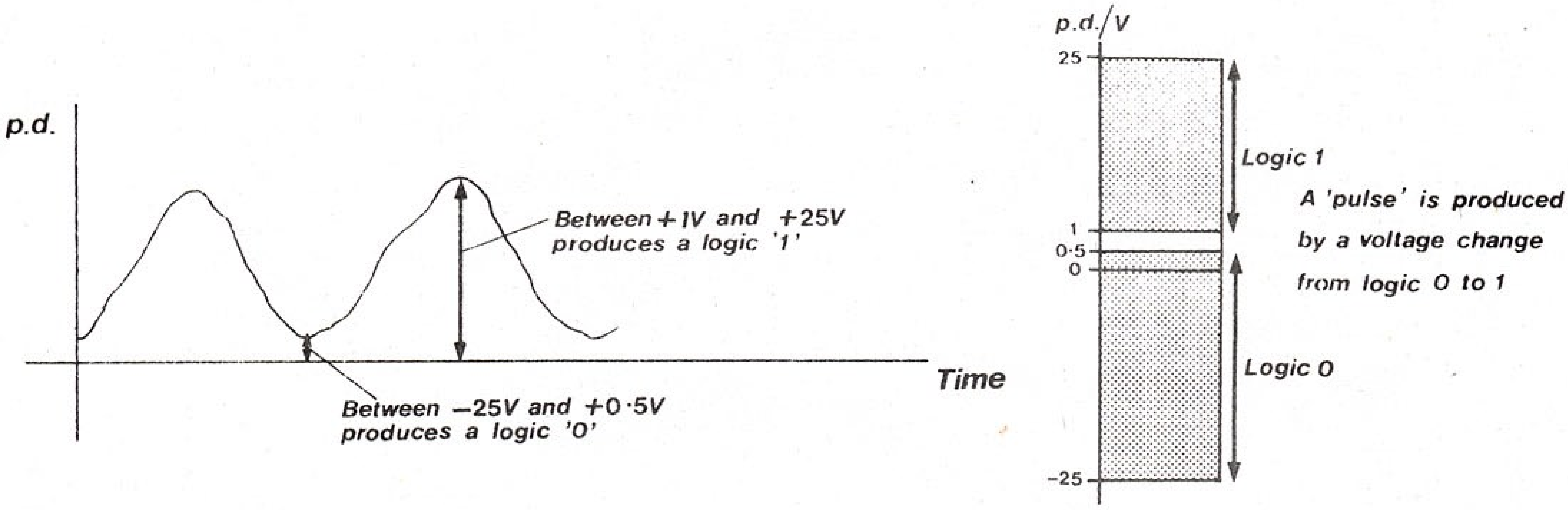
ROM = Read only memory
RAM = Random access memory
ADC = Analogue to digital converter
DAC = Digital to analogue converter
PIA = Peripheral interface adaptor

2 PULSE INPUT/TRIGGER INPUT

In some applications, data, which comes in the form of pulses or alternating waveforms, must be supplied to the instrument via the pulse input terminals, which are also on the left of the instrument next to the analogue inputs. (Examples of such data include timing pulses when using the instrument as a timer, and pulses which synchronise data logging with the event being monitored.)

The pulse input is connected to a pulse shaping circuit which changes state, or triggers, when the input exceeds approximately 1.0 V. Thus input pulses or waveforms of any shape can be connected to the pulse input providing that the "peak" voltage is greater than 1.0 V and the "trough" voltage is less than approximately 0.5 V, as shown in the diagram below. Therefore, the pulse shaping circuit introduces hysteresis so that 'clean', unambiguous pulse detection occurs even with relatively slowly changing signals which may have a certain amount of noise superimposed on them.

Note that the pulse input does not detect a zero crossing of the signal – merely a change from below approximately +0.5 V to above approximately +1 V (ie suitable for TTL level signals).


p.d.
Between +1V and +25V produces a logic '1'
Between −25V and +0·5V produces a logic '0'
Time
p.d./V
25
1
0·5
0
−25
Logic 1
Logic 0
A 'pulse' is produced by a voltage change from logic 0 to 1

The voltage limits on the pulse channel are +/–25 V. The light emitting diode (LED) on the top panel next to the input terminals will be illuminated when the pulse input is high, ie greater than 1.0 V. The input impedance of this circuit is approximately 1 M ohm. In some applications it is useful to be able to amplify a small amplitude input waveform (eg from a microphone) before the signal is fed to the pulse shaping circuit. The output from the channel 1 amplifier can be connected to the pulse input by means of the slider switch at the top left hand side of the front panel, underneath the LED. To use this facility, the pulses should be connected to channel 1. The channel 1 amplifer should be switched to the +/– 250 mV range. The pulse input slider switch should be switched to the right hand side linking it with the channel 1 input.

When using this facility, switching will occur much closer to a zero crossing condition, the switch thresholds now being approximately 100 mV and 50 mV respectively.

3 KEYPAD

This is pressure sensitive and occupies most of the top surface of the instrument. It is used for supplying instructions, to the instrument, as described in the Operating Instructions.

4 DISPLAY

The upper left hand corner of the top panel contains the 8-digit display. This is used, for example, to display the program number that has been requested and to display values of stored data. There are also 3 light emitting diodes in the display; these are used to indicate the appropriate units that accompany the numbers displayed (volts, seconds or hertz).

5 ANALOGUE OUTPUT

The analogue output socket is used to connect to an oscilloscope or chart recorder. This socket is situated on the right hand side of the instrument. The 4 mm sockets will take BNC adapters, RS stock number 456-009.

6 'SYNC' OUTPUT

This socket is on the right hand side of the instrument. It is for connecting to the 'external trigger' socket found on most oscilloscopes. Use of this facility gives better trace stability with most oscilloscopes. See Oscilloscope Instructions, for further details. The 4 mm sockets will take BNC adapters, RS stock number 456-009.

7 26-WAY DIGITAL SOCKET

This is on the right hand side of the instrument. To this socket are connected 16 digital input/output lines, together with 4 control lines. This socket is used:

a) for transfer of data to and from a microcomputer.

b) for monitoring up to eight 2-state sensors simultaneously (using for example the multiple timer program, no 06).

c) in connection with providing outputs of sequential codes for control applications.

Note that there is no overload protection on these lines. They are designed to accept or give out TTL compatible signals.

The pin connections to this socket are as follows:

1	Earth	14	PB0
2	Earth	15	PA7
3	+5 V	16	PA6
4	Earth	17	PA5
5	Control input/output (CB2)	18	PA4
6	Control input (CB1)	19	PA3
7	PB7	20	PA2
8	PB6	21	PA1
9	PB5	22	Control input/output (CA2)
10	PB4	23	PA0
11	PB3	24	Control input (CA1)
12	PB2	25	Earth
13	PB1	26	Control input (CA1)(ADDRESS $C002)

26-WAY DIGITAL SOCKET CONNECTIONS.

A suitable connector is RS stock No. 469 – 897

8 POWER SUPPLY

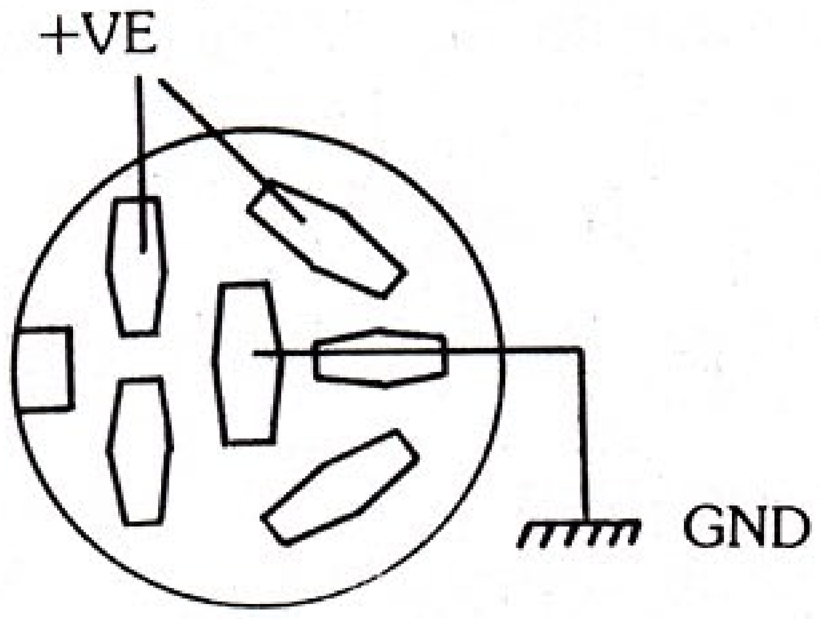
The Logger requires an 8 to 14 V DC or 7 to 9 V AC power supply, and draws a maximum current of 0.5 ampere. The power supply socket is on the rear of the instrument. A mains (240 V) power supply adaptor is included.

The unit can be used with an 8 to 12 V battery outside the laboratory (a battery of nominal voltage 12 V will give almost 14 V when fully charged). The size of the battery depends, of course, on the length of time for which the Logger will be operated; for example, a 6 hour run requires a battery capacity of 3 ampere-hours – easily provided by relatively small rechargeable cells.

POWER SUPPLY CONNECTIONS

The warning on the back of the instrument applies when it is NOT being used with the supplied mains adaptor. Do not use the supply which is powering the instrument to power external circuitry to which the logger may be connected as damage may occur.

It is however, permissible to use the +5 V output on the back of the logger to power sensors, etc, provided the current used is less then 70 mA.



View of power supply
connection from
rear of instrument

9 BATTERY BACK UP OF MEMORY

An onboard PCB battery provides data retention on power down. Data memory is provided by low standby power CMOS integrated circuits.

If the Logger has not been used for a considerable period of time (eg several weeks) then it should be left connected to a power supply for two days to ensure that the PCB battery is adequately recharged before use in the field. A trickle charging circuit for this battery is built into the instrument.

Data that is stored in memory after power down can be retrieved when power is reapplied by making use of the re-entry program number 08. Thus it can gather data in the field and later transfer it to a microcomputer, scope or chart recorder for further analysis. With a fully charged PCB battery, data will be retained in memory for at least a week.

OPERATING INSTRUCTIONS – GENERAL

These operating instructions must be read in conjunction with the instructions for the individual programs and the general description of the unit.

1 Connect the instrument to a suitable power source. A mains adaptor unit is supplied.

2 Connect the input sockets (ie the analogue and pulse inputs) on the left hand side to the equipment, or sensors, as appropriate.

3 Connect the output sockets on the right hand side to an oscilloscope, or other equipment, if required.

4 The keypad has been arranged as logically as possible. The left hand side is used to give instructions about which program is required, when to start and stop, etc. The right hand side is used to recover and display stored data, for example, after a data logging program.

5 Switch on the power. The word 'HELLO' will appear on the display for a few seconds, after which a 'P' will appear on the left hand side of the display.

6 Type the two-digit number of the program you wish to use on the keypad. Notice that all programs have two digits, so 'leading zeros' must be included for program numbers less than 10. (eg for program number 4, type '04'). The program number you type will appear at the left hand side of the display.

7 Many programs then require a 'parameter' to be typed in, for example to define the time between readings of the input voltage. This can be a one, two or three digit number. Type the required parameter on the keypad. This parameter will appear at the right hand side of the display as it is typed. Parameter details can be found in the particular program description section.

8 Check that the display is correct. If it is, press 'ENTER', if not press 'RESET' and start again. Note that if the program requested does not exist in any of the ROMs, the instrument will probably not respond to further entries. If this happens, press 'RESET', if a 'P' does not appear on the display, switch off the power and start again.

9 Those programs concerned with data logging and timing require a 'start' instruction. This can be provided by pressing 'START', or in many cases by providing a pulse to the pulse input socket. This is explained more fully in the appropriate instructions for the individual programs.

10 Many programs can be stopped by pressing 'STOP'.

OPERATING INSTRUCTIONS – FOR DATA LOGGING PROGRAMS

The right hand side of the keypad is used mainly to recover data stored during a data logging program, and control its display.

11 Data logging can be finished by pressing 'STOP' (except on program 01), or will stop automatically when the memory is full. The fact that data logging has finished is indicated by a flashing '0–P' on the display.

12 After logging is complete, press 'CH 1', 'CH 2', 'CH 3' or 'CH 4' according to which channel was used to log the data. The chosen channel number will appear on the left hand side of the display.

13 Press 'SCOPE', 'CHART' or 'MICRO' according to whether the data is to be read out on an oscilloscope, a chart recorder or transferred to a microcomputer. It is necessary to correctly set up the relevant piece of equipment to receive the data before pressing one of these keys.

14 Data is sent to an oscilloscope repetitively; while data is being sent to an oscilloscope, the chosen channel number is shown on the left of the display, and the value of the first item of data logged on that channel appears on the right of the display. To stop data output to an oscilloscope press 'RESELECT DISPLAY'; a flashing '0-P' will appear on the display; return to instruction 12.
FOR PROGRAM 01 ONLY:- 1K of data is sent to the oscilloscope at any one time; pressing 'CH 2', 'CH 3' or 'CH 4' selects the second, third or fourth 1K data block respectively for display.

FOR PROGRAMS 02 AND 03 ONLY:- (medium and slow speed transient recorders) If channel 1 is selected, only data stored in that channel is sent to the oscilloscope.

15 Data is sent to a chart recorder or microcomputer only once; after data transfer is complete, the display will show a flashing '0-P'; return to instruction 12 or re-commence logging.

16 Data can be shown one item at a time on the display as follows:

a) Choose the channel number as in 12 above. The number of the selected channel will be shown on the left of the display.

b) Press 'SCOPE' as in 13 above, (even if there is no oscilloscope connected). The chosen channel number will appear on the left of the display, a '1' (for 'first item') will appear in the middle of the display, and the value of the first item of data logged on the chosen channel will appear on the right of the display as described in 14 above.

c) Press ' > '; the chosen channel number will flash momentarily on the left of the display, the 'item number' (2) will appear in the middle of the display, and the second item of data will be shown on the right. Press ' > ' again and the third item of data will be shown, and so on.

d) To move backwards through the data, use the ' < ' key instead of the ' > ' key.

e) A bright up cursor is displayed on the oscilloscope corresponding to the position of the data currently on the logger's display.

f) If the '>' key is pressed at the same time as the '<<>>' key, the cursor will move forwards at a quicker rate through the waveform. The cursor will continue to move until pressure on the '<<>>' keypad is released.

17 To change the data channel, or to change the instrument onto which the data is transferred (eg from oscilloscope to chart recorder), press "RESELECT DISPLAY" and then start again at instruction 12.

18 In order to run the program again with exactly the same parameter press "RESELECT" DISPLAY" followed by "ENTER". The logger will wait for a suitable start pulse or until the "START" key is pressed.

19 **IN THE EVENT OF PROBLEMS:**

"HELLO" should be indicated when the unit is switched on. If this does not occur then please check:

a) that the fuse on the rear panel has not blown. If this needs replacing, a fast blow 1 A fuse should be used.

b) that the power supply being used is capable of delivering 0.5 A at a minimum of 8 V ac or dc. This only applies when the mains adaptor supplied is not being used.

If the logger fails to operate as expected or does not respond in a predictable manner to instructions typed in from the keypad it is suggested that you check the following:

a) Check that transients on the mains (easily caused by plugging or unplugging items of equipment into the mains near the power supply) are not causing the programs to 'crash'. Like most other microprocessor based pieces of equipment, the logger is sensitive to spikes on the mains and whilst this is unlikely to cause any permanent damage, it will lead to apparent operating malfunctions. If this is the case, turn off the power supply, power up again, re-enter the program and proceed as before.

b) Look at the waveform of the power supply to the unit with an oscilloscope to check that it is satisfactory. Some power supplies incorporate thyristor switching and are unsatisfactory as the very brief switching transients can interfere with microprocessor circuitry.

OSCILLOSCOPE INSTRUCTIONS

USING AN OSCILLOSCOPE TO DISPLAY THE DATA

With the data logging programs it is possible to connect an oscilloscope to the analogue output to provide a means of displaying the captured data. With program 03 the data is available on the analogue output line as it is read from the experiment, so that an oscilloscope can build up a graph of the data as it is logged. Otherwise the data is available on the analogue output line after all the logging is complete. The following notes are to help you obtain a steady, clear trace on your oscilloscope as quickly as possible.

CONNECTIONS TO OSCILLOSCOPE:-

An analogue output socket is provided on the right hand side of the instrument, for connecting to an oscilloscope. If trouble is experienced with picking up mains hum on the oscilloscope, check that the connecting leads are routed well away from any mains cables. If necessary, use a coaxial connecting lead. Note that the analogue output socket will accept a 4 mm to BNC socket. Adaptor stock number 456–009

Y-SENSITIVITY:-

The output has maximum values of +/–2.5 V. If there are 10 divisions in the y-direction on your oscilloscope screen, (as is typical), then a sensitivity of 0.5 V/divison will be found to be suitable for virtually all applications.

TIMEBASE SPEED:-

It takes about 0.05 milliseconds to output each item of data to the oscilloscope. The timebase that you use depends on how much data you wish to display on the oscilloscope. For example, the slow speed transient recorder can collect a maximum of 1023 readings per channel, and it will take just over 50 milliseconds to output them all to an oscilloscope. To display all these readings will require a timebase speed of 5 milliseconds/division (assuming the x-axis is divided into 10 divisions).

If fewer readings are required to be displayed then a faster timebase speed can be used, eg a timebase speed of 1 milliseconds/division will display approximately the first 200 readings.

If in doubt (eg you are not sure how many relevant readings you have stored) it is suggested you start with a timebase speed of 5 milliseconds/division, and then adjust it as necessary.

Notice that you have to start reading from the first item of data. You cannot, for example, just display the last 100 items of data.

TRIGGER LEVEL:

This is the key control for a stable oscilloscope trace.

Oscilloscopes have a trigger facility to make the trace on the screen start when the input voltage at the Y-input reaches a preset level. This is done so that an oscilloscope can always start its trace at the same point on an incoming signal, thus ensuring a stable trace. The trigger control can usually be set to ‘automatic’, in which case the trace will start when the input voltage is zero (eg midway between the positive and negative peaks of an ac waveform).

However, to display the captured data, the oscilloscope needs to start its trace at the start of the data (which will not usually be zero!). So that the oscilloscope ‘knows’ where the start of the data is, a short +2.5 V pulse is produced just before starting to send out the data. If the trigger level control is set to approximately 2.5 V, then the oscilloscope trace will start as soon as that pulse arrives, and hence you should have a stable trace.

In practice the trigger level control is not calibrated, so you have to find the right setting of the control by trial and error. The important thing is that the control should NOT be on its ‘automatic’ setting; you must set the trigger level yourself.

EXTERNAL TRIGGERING:

Even with the above internal trigger control working properly, it is still sometimes difficult to obtain a stable trace. This particularly occurs if the output signal to the oscilloscope rises to near +2.5 V, when the oscilloscope can become confused between the output signal and the trigger pulse.

To overcome this problem, most oscilloscopes are provided with an 'external trigger' socket. The oscilloscope trace will trigger when a pulse arrives at this socket, provided the appropriate control on the oscilloscope has been switched to 'external trigger'.

There is an 'external sync' socket on the right hand side the unit. A pulse is given out from this socket at the start of the waveform display. To use this facility, connect a lead between the 'external sync' sockets on the logger and oscilloscope, and switch the appropriate trigger control on the oscilloscope to 'external'.

CHART RECORDER INSTRUCTIONS

Output of data to a chart recorder.

With most of the data logging programs it is possible to connect a chart recorder to the instrument to provide a 'hard copy' of the data stored. The chart recorder should be connected to the analogue output on the right hand side of the unit (the same output that is used for an oscilloscope).

Y-SENSITIVITY: The maximum output range is +/–2.5 V. The y-sensitivity of the chart recorder should be set so that full scale deflection is obtained with a voltage greater or equal to +/–2.5 V. Notice that the chart recorder pen must be set to the middle of the paper if the chart recorder is to respond to negative values of voltage.

TIMEBASE SPEED:

Data is transferred to the chart recorder at the rate of 1023 items in approximately 5 minutes. A timebase speed of 10 cm/minute will be found suitable in nearly all cases. The time axis can be calibrated if the rate at which data was originally collected is known.

PROCEDURE:

Connect the chart recorder to the Data Logger's output, and switch it on. Data transfer to the chart recorder is started by selecting a data channel and then pressing 'CHART' on the right hand side of the keypad; see the individual program instructions for further details. Switch on the chart recorder motor BEFORE pressing 'CHART'.

The data is sent to the chart recorder in the reverse order to that in which it was collected, ie last item first, so that the resulting graph is the 'right way round'. After it has sent all the data, a y–axis will be drawn on the chart. Disconnect the chart recorder after this has occurred, then press 'RESELECT DISPLAY' on the keypad.

NOTE

It takes about 5 minutes to output all the data stored in one channel to a chart recorder. This slow speed enables the majority of chart recorders to respond to the fine detail in the captured waveforms.

MICROCOMPUTER INSTRUCTIONS

Before data can be passed to a microcomputer a suitable cable and transfer software in the microcomputer (to accept the incoming data and provide the correct communication protocols) have to be available.

A suitable software package and cable are available for the BBC B microcomputer – RS Stock Number 611-997

Data transfer is in parallel format to computer user I/O ports.

The steps in data transfer are as follows:-

a) Connect the microcomputer to the digital I/O socket on the side of the logger.

b) Load and run the appropriate microcomputer program.

c) Press 'CH 1', 'CH 2' (etc according to which channel or block of data is to be transferred).

d) Press 'MICRO'.

e) When the data transfer is complete, the display will show 'O-P' and a new data block or output instrument can be chosen.

SUMMARY OF AVAILABLE PROGRAMS IN STANDARD ROM

Program Number		Parameter
00	Four channel digital voltmeter	1 to 4 (channel number)
01	Fast transient recorder (single channel)	0 to 999 (x50 microseconds)
02	Analogue (transient) recorder	1 to 999 (milliseconds)
03	Analogue recorder, slow	1 to 999 (seconds)
04	Frequency meter	
05	Event timer	1 to 4 (pulse type)
06	Multichannel timer	
07	Unassigned	
08	Re–entry to output routines	
09	Download from microcomputer	
10	Waveform generator	0 to 999 (milliseconds)
11	Control sequence generator	1 to 999 (seconds)
12	User program creation	

SUMMARY OF AVAILABLE PROGRAMS IN EXTENDED FACILITIES ROM

20	4 channel sampling by microcomputer with real time data transfer
21) 22) 23)	Data table to printer of data logging programs
24) 25) 26)	Graphics output to printer of data logging programs
27	4 Channel Real time logger to printer
28	Data table to printer of multi-channel timer

FOUR CHANNEL DIGITAL VOLTMETER - PROGRAM NUMBER 00

Description: This program measures the voltages at the four analogue inputs and shows the value of one of these voltages on the display. Its primary purpose is to check sensors, circuits, etc prior to data logging.

Input: Any dc voltage in range +/–25 V to each of the four analogue inputs. The inputs share a common ground.

To use this program:

1 Connect the circuit, sensors, etc to the analogue inputs as appropriate;

2 select the appropriate input voltage range using the three position slider switches;
3 press 'RESET';

4 press '00' to select this program;

5 press 'ENTER';

6 the right hand side of the display will now show the input voltage (in volts) of channel 1. The display is updated every half second.

7 To change the channel which is being read, press 'CH1', 'CH 2', 'CH 3' or 'CH 4' as appropriate. The left hand side of the display shows the channel which is being monitored.

8 If the input is too high for the range selected, the display flashed 'HI'. If it is too low it flashes 'LO'.

FAST TRANSIENT RECORDER – PROGRAM NUMBER 01

Description:

This program records the voltage at the 'channel 1' analogue input as a function of time. The sample rate can be 26 microseconds, or multiples of 50 microseconds up to 999 x 50 microseconds.

Parameters:

0 to 999. This defines the time. A parameter of 0 will give 26 microseconds between readings. A parameter between 1 and 999 gives 50 microseconds multiplied by that parameter. For example, a parameter of 1 instructs the logger to take readings every 50 microseconds; a parameter of 2 means readings every 100 microseconds; and so on.

Input:

To channel 1. The input must be in the range +/–25 V.

A pulse might also be required on the pulse input to start data logging. (See below).

Maximum data:

4092 readings.

To use this programe:

1 Make appropriate connections to the channel 1 input;

2 select the appropriate input range using slider switch (+/–25 V, +/–2.5 V or +/–250 mV);

3 press 'RESET';

4 press '01' to select this program;

5 use the keypad to type in the parameter, ie to define the time between readings as explained above;

6 press 'ENTER'; the parameter will disappear from the display;

7 to actually start recording data, EITHER

a) press 'START' or

b) apply a positive pulse (greater than 1 V) to the pulse input – the pulse switch could be set to receive a signal from channel 1 ensuring automatic starting when the input waveform exceeds the trigger threshold. This enables the data collection to be synchronised with the event being monitored. A delay of approximately 0.7ms occurs between receiving a pulse and the start of logging.

When 'START' is pressed, the 'sync' output on the right goes from low to high (about 4 V), and stays high for the duration of the data logging. This high output can be used to start an experiment running, and provides an alternative means of synchronising the data logging with the experiment.

To stop logging data:

Logging will stop automatically after 4096 items of data have been recorded. The 'STOP' keypad will have no effect in this program.

Logging finished:

The centre of the display shows a flashing 'O-P' when data logging is finished.

Output:

The output instructions are given using the right hand half of the keypad. The data is stored in memory in four blocks of 1023 readings.

Readings 1 to 1023 are in block 1,
1025 to 2047 are in block 2,
2049 to 3071 are in block 3,
3073 to 4096 are in block 4.
(Readings 1024, 2048 and 3072 do not exist)

NOTE: The last 7 bytes of block 4 do not contain data but program parameters.

In any of the output methods previously described, only ONE block of data can be handled at a time.

In order to run the program again with exactly the same parameter press "RESELECT DISPLAY" followed by "ENTER". The logger will wait for a suitable start pulse or until the "START" key is pressed.

ANALOGUE (TRANSIENT) RECORDER - PROGRAM NUMBER 02

Description: This program records the voltage at all four input channels simultaneously as a function of time. The time between readings is defined by a parameter typed in by the user. All Channels share a common ground.

Parameter: 1 to 999. This defines the time in milliseconds between each reading of the input.

Input: On any combination of the four channels. The input to each channel must be in the range +/–25 V.

Maximum data: 1023 readings per channel.

To use this program:

1 Make appropriate connections to the inputs;

2 select the appropriate input voltage range to each channel using the slider switches;

3 press 'RESET'

4 press '02' to select this program;

5 using the keypad, type in the time (in milliseconds) between readings (the 'parameter');

6 press 'ENTER'; the 'parameter' disappears from the display;

7 to actually start recording data, EITHER

a) press 'START', or

b) apply a positive pulse (greater than 1 V) to the pulse input.

This enables logging to be synchronised with the event being monitored. The pulse switch could be moved to the right so that logging starts automatically when the voltage on Channel 1 exceeds the trigger threshold.

When 'START' is pressed, the 'sync' output on the right hand side goes from low to high (about 4 volts), and stays high for the duration of the data logger. This high output can be used to start an experiment running, and provides an alternative means of synchronising the data logging with the experiment being monitored.

8 to show that logging is taking place the 'seconds' light on the display will flash on and off.

To stop logging data: press 'STOP'. (Logging will stop automatically after 1023 items of data per channel have been recorded.)

Logging finished: This is indicated by '0-P' on the display.

Output: Refer to the output methods described previously.

In order to run the program again with exactly the same parameter press "RESELECT DISPLAY" followed by "ENTER". The logger will wait for a suitable start pulse or until the "START" key is pressed.

The user should note that the 4 channels are measured sequentially and not simultaneously. The total time for all 4 measurements is small compared with the shortest intersample time of 1 mS available with this program.

Section 2.3

ANALOGUE RECORDER (SLOW) – PROGRAM NUMBER 03

Description:

This program functions in the same way as program 02 and records the voltage at all four input channels simultaneously as a function of time. The time between readings is defined by a parameter typed in by the user, and is in the range 1 to 999 s. All channels share a common ground.

Parameters:

1 to 999. This defines the time in seconds between each reading of the input.

Input:

To any combination of the four channels. The input to each channel must be in the range +/–25 V.

Maximum data:

1023 readings per channel.

To use this program:

1 Make appropriate connections to the inputs;

2 select the appropriate input range to each channel using the slider switches;

3 press 'RESET';

4 press '03' to select the program number;

5 using the keypad, type in the time (in seconds) between readings (the 'parameter');

6 press 'ENTER'; the parameter disappears from the display;

7 to actually start recording data, EITHER

a) press 'START' or

b) apply a positive pulse to the pulse input. This enables logging to be synchronised with the event being monitored. The pulse switch could be moved to the right so that logging starts automatically when the voltage on channel 1 exceeds the triggering threshold.

8 When data logging starts, an oscilloscope connected to the analogue output displays an updated graph of the value of the channel 1 input against time. Press 'CH 2' to display the data on channel 2, 'CH 3' to display the data on channel 3, etc.

The display shows the chosen channel number, the number of items of data that have been logged on that channel and the value of the voltage presently being monitored on that channel.

To stop logging data:

press 'STOP'. (Logging will stop automatically after 1023 items of data per channel have been recorder.)

Logging finished:

This is indicated by 'O-P' on the display.

Output:

Refer to output methods described previously.

In order to run the program again with exactly the same parameter press "RESELECT DISPLAY" followed by "ENTER". The logger will wait for a suitable start pulse or until the "START" key is pressed.

The 4 channels are measured sequentially and not simultaneously but the period over which the measurements are taken is short compared with the shortest intersample time of 1 second for this program. This program includes signal averaging to reduce any noise present on the input signals: 256 readings are taken of each channel and the average reading from each channel stored in memory.

FREQUENCY METER – PROGRAM 04

Description:

This program measures the frequency of pulses, or of a waveform, which can be of any regular shape. It is only suitable for use with frequencies < 20 KHz. Its primary purpose is to measure the frequencies of waveforms prior to data logging to enable the user to select appropriate intersample times.

Parameter:

None

Input:

To the pulse input. Input should be in the range +/–25 V. The signal should have a peak amplitude of at least +1 V. Signals of less than +1 V peak amplitude will first need amplifying, by connecting to the channel 1 input, selecting a suitable range with the three position range switch and switching the pulse input to 'internal'.

Frequency range:

1Hz to 20KHz. The display flashes 'HI' if the input frequency is of a suitable amplitude but too high a frequency to be measured and has been increased beyond approximately 20 KHz.

To use this program:

1 Make appropriate connections to the pulse input if necessary via the channel 1 amplifier (see above);

2 press 'RESET';

3 press '04' to select the program number;

4 press 'ENTER'.

Output:

The frequency is displayed, in Hz, on the display. This is updated once every second.

NOTE: The display shows the last non–zero count in any previous interval.

EVENT TIMER / STOPWATCH – PROGRAM NUMBER 05

Description: This program records the time between 'start' and 'stop' signals which can be sent either via the pulse input or from the 'START' and 'STOP' keys.

Parameter: 1, 2, 3 or 4. The default value is 1. This defines the kind of pulse which will start and stop the clock. See accompanying diagram.

Input: To the pulse input. The pulse must be in the range +/–25 V and should have a peak amplitude of at least +1 V. Signals of less then +1 V peak amplitude can be amplified by connecting them to the channel 1 input, and selecting a suitable range with the three position range switch. In this case the pulse input switch should be on 'internal', thus connecting the output from channel 1 amplifier to the pulse input.

Time range: 1 millisecond to 65 seconds.

To use this program:

1 Make appropriate connections to the pulse input, if necessary via the channel 1 amplifier (see above);

2 Press 'RESET';

3 Type '05' to select this program;

4 Press a number in the range 1 to 4 to select the appropriate parameter (see below);

5 Press 'ENTER';

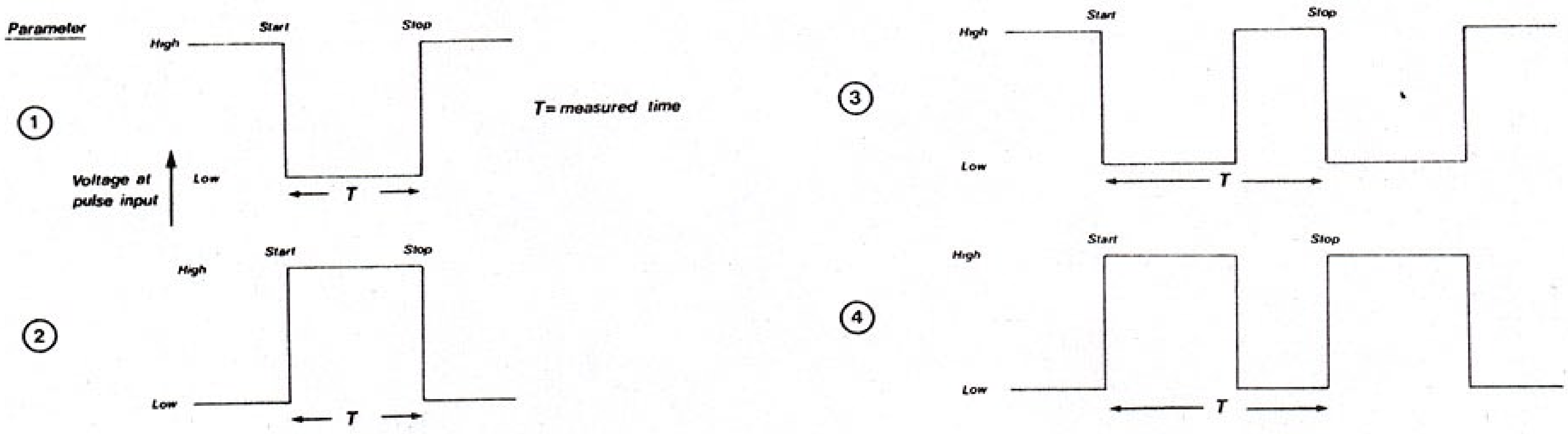
6 Timing will start when the 'START' key is pressed or when a starting pulse is received, and stop when the 'STOP' key is pressed or a stop pulse is received; the 'seconds' light on the display will flash at approximately 2Hz while timing is in progress;

7 There is no need to reset the timer for subsequent timings unless a different kind of start or stop pulse is required. There is a delay of 1 second before the logger can start timing again.

Output: The time is displayed, in seconds, on the display. If the time exceeded 65 seconds, the logger 'times out'; the display shows 'HI'; in this case it is necessary to press 'RESET' and start again.

Start and stop pulses: The parameter (1 to 4) defines whether the unit starts or stops timing as the input voltage goes from low to high (a positive going edge'), or as the input goes from high to low ('negative going edge'), as in the diagrams below.

Regardless of the parameter, the 'START' key can be used to start the timing, and the 'STOP' key to stop the timing.



MULTI-CHANNEL TIMER – PROGRAM NUMBER 06

Description:

This program monitors up to 8 sensors and records.

a) The times at which any of the sensors changes state;

b) Which of the 8 sensors changes.

The sensors must give TTL compatible logic level signals.

Parameter:

None

Input:

To 8 digital input lines, (lines PA0 to PA7), one sensor output being connected to each digital input. The input must be a TTL compatible logic signal as described above. Note that the digital input lines are not overload protected.

Time range:

1 millisecond to 65 seconds.

To use this program:

1 Connect each sensor to a separate digital input line;

2 press 'RESET';

3 type '06' to select this program;

4 press 'ENTER'; the display will show the state of each of the input lines; for example, if each sensor is sending a 'high' signal to the input lines, the display will read 11111111;

5 press 'START' to start timing; the display clears and the seconds light will flash while timing is in progress;

6 to stop timing press 'STOP'; the program recycles from zero after 65.536 seconds.

Output:

1 On the display

a) When timing has finished, press '>'. The display will show a series of eight digits, either 1 or 0, which shows the state of the sensors when timing started; the first digit shows the state of line PA7 ('most significant bit') and the last digit the state of line PA0 ('least significant bit').

For example, if the display shows 11111101, then all sensors were giving a 'high' output except sensor number 2 (connected to line PA1) which was giving a 'low' output.

b) Press '>' again. The display will show 'O'; this is the time (in seconds) at which the above 'sensor pattern' occurred, ie at the start

c) Press '>' again. The display will show the next 'sensor pattern' which occurred. eg 11111111, all sensors were giving a high output.

d) Press '>' for a fourth time. The display will show the time, in seconds, at which the sensor outputs changed to this pattern.

e) Pressing '>' further times gives successive sensor patterns and the times at which the sensors changed to that pattern.

f) After displaying the time at which the last change occurred, a further press of '>' returns the display to the start of the sequence again.

g) Note that '<' does not operate with this program.

2 Transfer of data to a microcomputer

It is necessary to program the microcomputer to receive the data. See the Technical Section of this manual.

a) Connect the microcomputer to the digital output socket on the right hand side.

b) Load and run the appropriate microcomputer program.

c) Now refer to program 08.

d) the first byte of data is the decimal equivalent of the binary number which represents the state of the sensors (eg a number 3 represents a sensor pattern 00000011); the next two bytes represent the time at which this sensor pattern occurred.

e) when data transfer is complete, press 'RESELECT DISPLAY'.

The shortest time period resolved with this program is 1 millisecond.

RE-ENTRY TO OUTPUT ROUTINES – PROGRAM NUMBER 08

Description: This program allows re-entry to the '0-P' stage of the other programs. It can be used after power down when re-powered again and also if the RESET key has been pressed accidentally.

Parameter: The same as the program number it is desired to re-enter.

To use this program:

1 When re-applying power the instrument automatically goes to the 'P' prompt. Type in 08 followed by the program number last in use.

2 This automatically returns to the 0-P indication on the display and then 'CHART', 'SCOPE' or 'MICRO' can be pressed as usual.

3 Although this program's primary use is in programs 01, 02, and 03 it can be used in 06 to return to the start of the multi-channel timer data sequence and to recover from pressing the RESET key in program 10.

DOWNLOAD DATA/PROGRAM CODES FROM MICROCOMPUTER – PROGRAM NUMBER 09

Description: Data and program codes can be transferred from a host microcomputer to the logger.

Uses include synthesis of waveform data tables on a host microcomputer prior to output via the logger's D to A converter and downloading of user created programs.

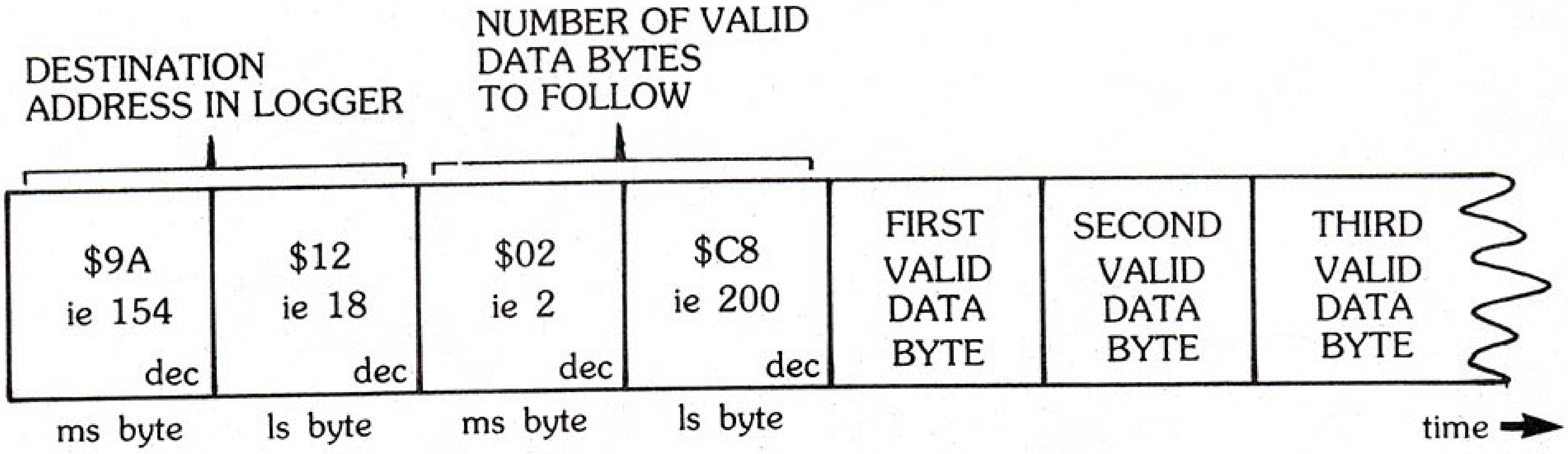
Parameter: None

To use this program: The keypress sequence:

0 9 ENTER

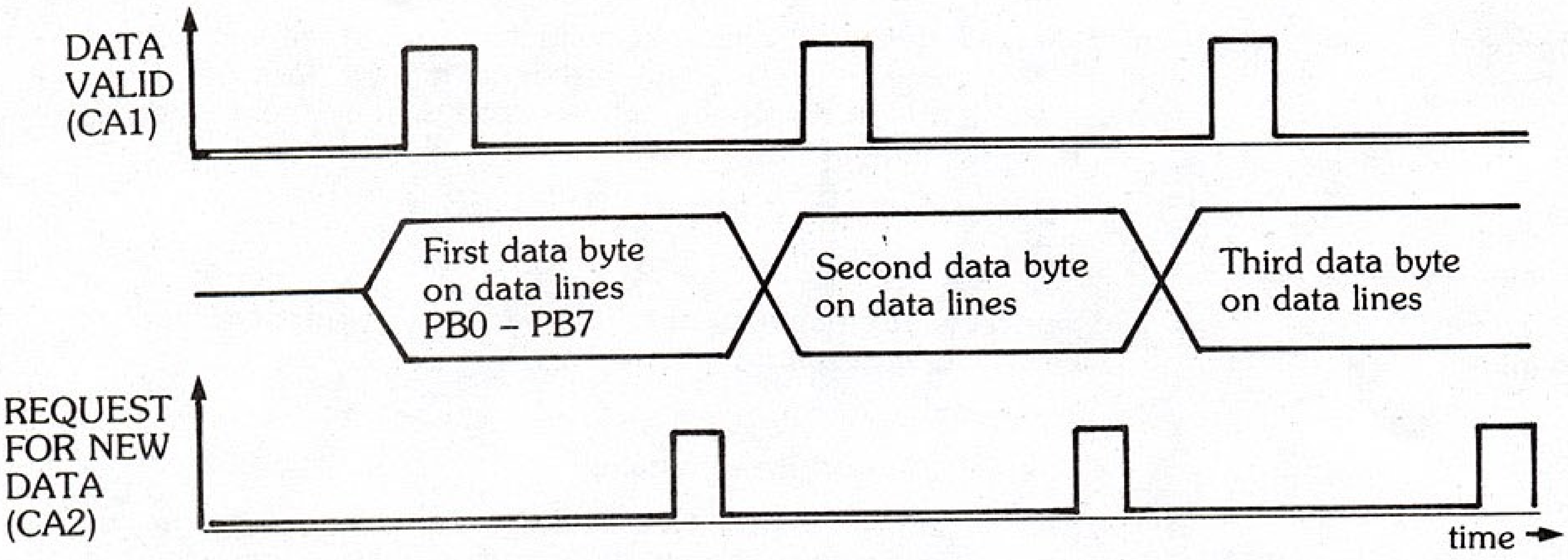
will put the logger into a standby mode, waiting for a positive going pulse on the CA1 control line (digital I/O port pin 24) to initiate the transfer of data or program codes. The cable linking the logger to the host microcomputer should have 8 data lines, 2 control lines and ground. (See technical section of this manual.) A suitable cable for the BBC microcomputer is supplied with the RS data logger analysis software package.

The instrument assumes that the first four bytes give the destination address of the first valid data byte, and also the total number of valid data bytes to be transferred, eg if the destination address is $9A12 and a total of 712 valid data bytes are to be downloaded, the first four codes will be:



PARALLEL TRANSFER PROTOCOL

The time sequence of operations is illustrated below. Note that it is simply the 'transfer data to microcomputer' handshake reversed (see technical section of this manual). The control line, CA1 is now used as the 'data valid' pulse line and control line, CA2 is the 'new request for data' pulse line.



Taking a specific example of a program which is to be downloaded from a BBC microcomputer. Suppose the program to be run is given by:

```
'again' INC $E001       7C E001  124,224,1
        JSR $F9F9       BD F9F9  189,249,249
        BRA 'again'     20 F8    32,248
```

(Which will make the volts, seconds, Hz LED's count slowly in binary.)

Suppose also that we want to send the first OP–Code (7C) to memory location $8C01 so that it can be tested by entering Program '12' – the User Creation Program.

The following basic program will achieve this transfer.

CASE STUDY: BBC MICROCOMPUTER DOWNLOADER ROUTINE

```
 100   REM A SIMPLE DOWNLOADER
 110   NIFR=&FE6D:PCR=&FE6C
 115   DDRB=&FE62:IER=&FE6E
 120   ?NIFR=&FF:?PCR=&F0
 140   IF INKEY(10)=-1 THEN 140
 200   PROC transfer
 250   END
2400   DEF PROCtransfer
2460   ?DDRB=&FF:?IER=0:?NIFR=&FF:?PCR=&FO
 250   FOR M=1 TO 12
2505   READ N%
2510   ?&FE60=N%
2520   ?PCR=&DO:?PCR=&F0
2530   DR%=?NIFR
2540   IF DR%=0 THEN GOTO 2530
2550   ?NIFR=&FF
2570   NEXT M
2580   REM End of Transfer
2600   ENDPROC
2650   DATA 140, 1, 0, 8, 124, 224, 1, 189, 249, 249, 32, 248
```

After creating this program on the BBC microcomputer, connect the BBC to the logger.

Press 0 9 and ENTER

At this stage the instrument is waiting for a transfer. Press any key on the BBC Micro and the transfer of the twelve data values will take place. The instrument should now display 'P-' to indicate that the transfer is finished.

Press 1 2 and ENTER and the last 8 data values should be in the first 8 memory locations. Press START and the logger will execute the machine code program – you should see the LED's cycling slowly.

VERSATILE WAVEFORM GENERATOR – PROGRAM NUMBER 10

Description: Using this program a user can build up a waveform of complex shape and duration. The waveform is available in analogue output, and in digital form from the eight digital output lines.

Parameter: 0 to 999. This defines the time in milliseconds between each code output. A parameter of 0 enables the instrument to work with approximately 80 microseconds between codes. If no parameter is typed, a default time of 1 ms between codes will result.

Input: From the keypad; numbers between 0 and 255.

Maximum data: 256 steps.

To use this program:

1 Connect your equipment or oscilloscope to the output socket or digital output as appropriate;

2 Press 'RESET';

3 Type '10' to select this program;

4 Use the number pad to type in the parameter, ie the time in milliseconds between the output of each code (see below);

5 Press 'ENTER';

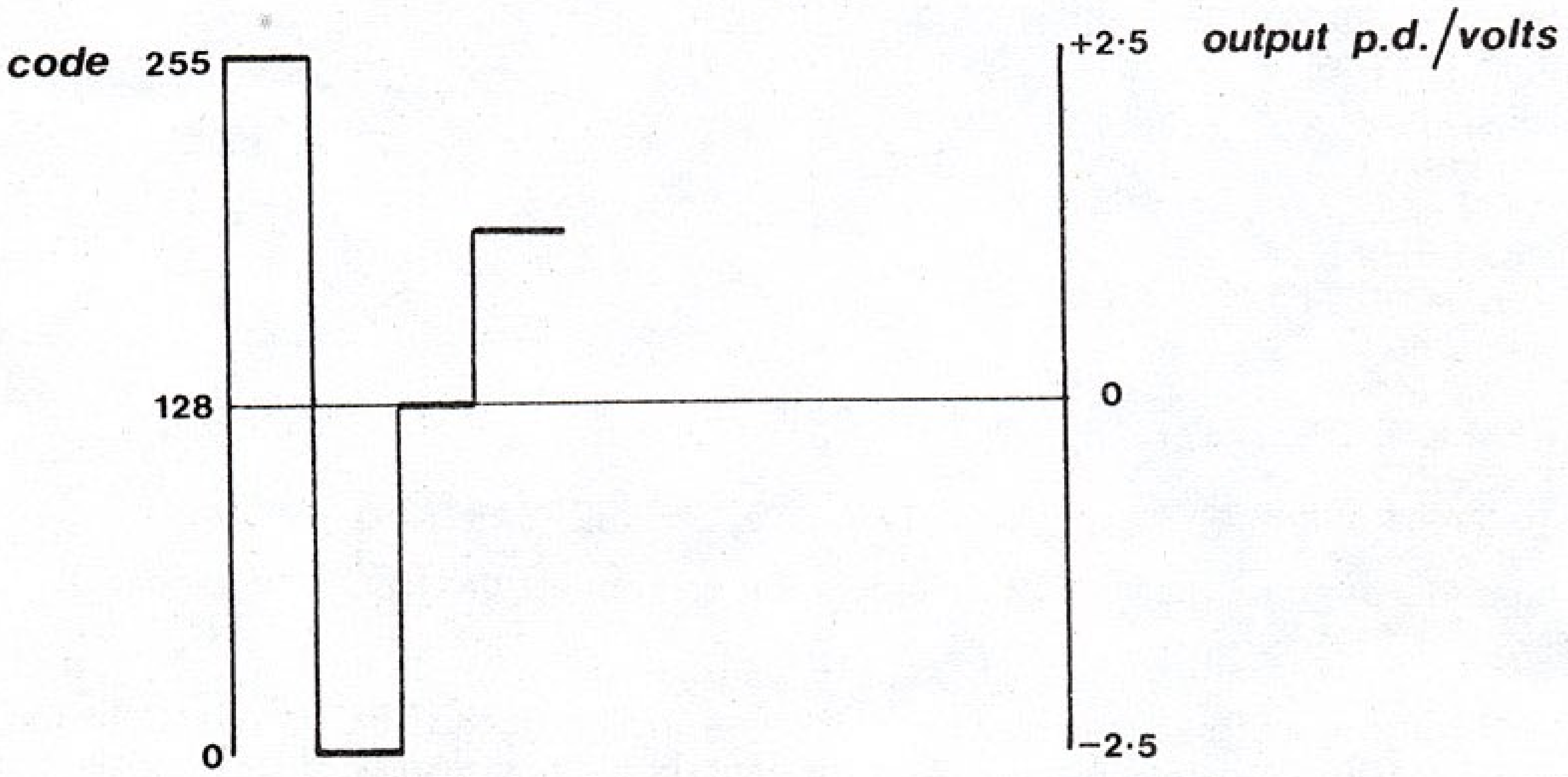
6 The display will now show '1' in the middle, and a code 'xyz' on the right, meaning that the contents of memory location 1 is 'xyz'; use the keypad to type in the code (between 0 and 255) that is required to replace xyz;

7 Press 'ENTER'; your code is now entered into location 1;

8 A voltage proportional to the code will appear at the analogue output, and the binary form of that number will appear on the digital output lines. The current output capability of the analogue output socket is limited by 600 ohm impedance.

For example, a code of 255 will give 2.5 V at the analogue output, and all the digital lines will go high, a code of 0 will give –2.5 V and all digital lines low; a code of 128 will give approximately 0 V and the state of the digital lines will be 10000000; a code of 192 will give 1.25 V and the state of the digital lines will be 11000000, and so on. See the diagram below.

At the same time, one or more of the three leds in the display may come on; these leds will reflect the three most significant bits of the code.



9 Press '>' to move to the next memory location, and enter the code you require in that memory location; '<' can be used to move back through memory in the same way.

10 As the codes are entered, an oscilloscope connected to the output displays the waveform that is being built up; the oscilloscope screen will look similar to the diagram above;

11 After entering all the required codes, press 'START'; the codes will appear repetitively on both the analogue output and the digital output lines; the time for which each code is present on these output lines is determined by the parameter entered at stage 4 above.

NB It is assumed that the last code entered is the end of the sequence; if you go back to make an alteration, you must then step forwards again to the end of the sequence before pressing 'START'; the last code entered remains on the display;

12 To alter any codes,

a) press 'RESELECT DISPLAY'; the output stops;

b) press 'SCOPE';

c) press '>' or '<' as appropriate to move to the relevant code;

d) alter the code as explained above;

e) after making all the required alterations, use the '>' key to step through to the final code before pressing 'START' again;

13 To change the time between the output of each code,

a) press 'RESELECT DISPLAY'; the output stops;

b) type the new time required;

c) press 'ENTER'; the output will start again automatically.

Output: At the analogue output and in binary coded form on the digital lines as described above.

NOTE

If 'RESET' is pressed in error, the codes remain in the memory. The codes can be recovered by pressing 08 followed by a parameter of 10 and then ENTER.

CONTROL SEQUENCE GENERATOR – PROGRAM NUMBER 11

Description: This program works in a similar way to the previous one – (Versatile Waveform generator) but the time between codes is longer and the code sequence is given out only once, rather than repetitively. Using this program a user can build up a sequence of codes, which can then be given out in analogue form at the analogue output, and in digital form from the eight digital output lines.

Parameter: 1 to 999. This defines the time in seconds between each code output. If no parameter is typed, the logger will default to 1 s between codes.

Input: From the keypad; numbers between 0 and 255.

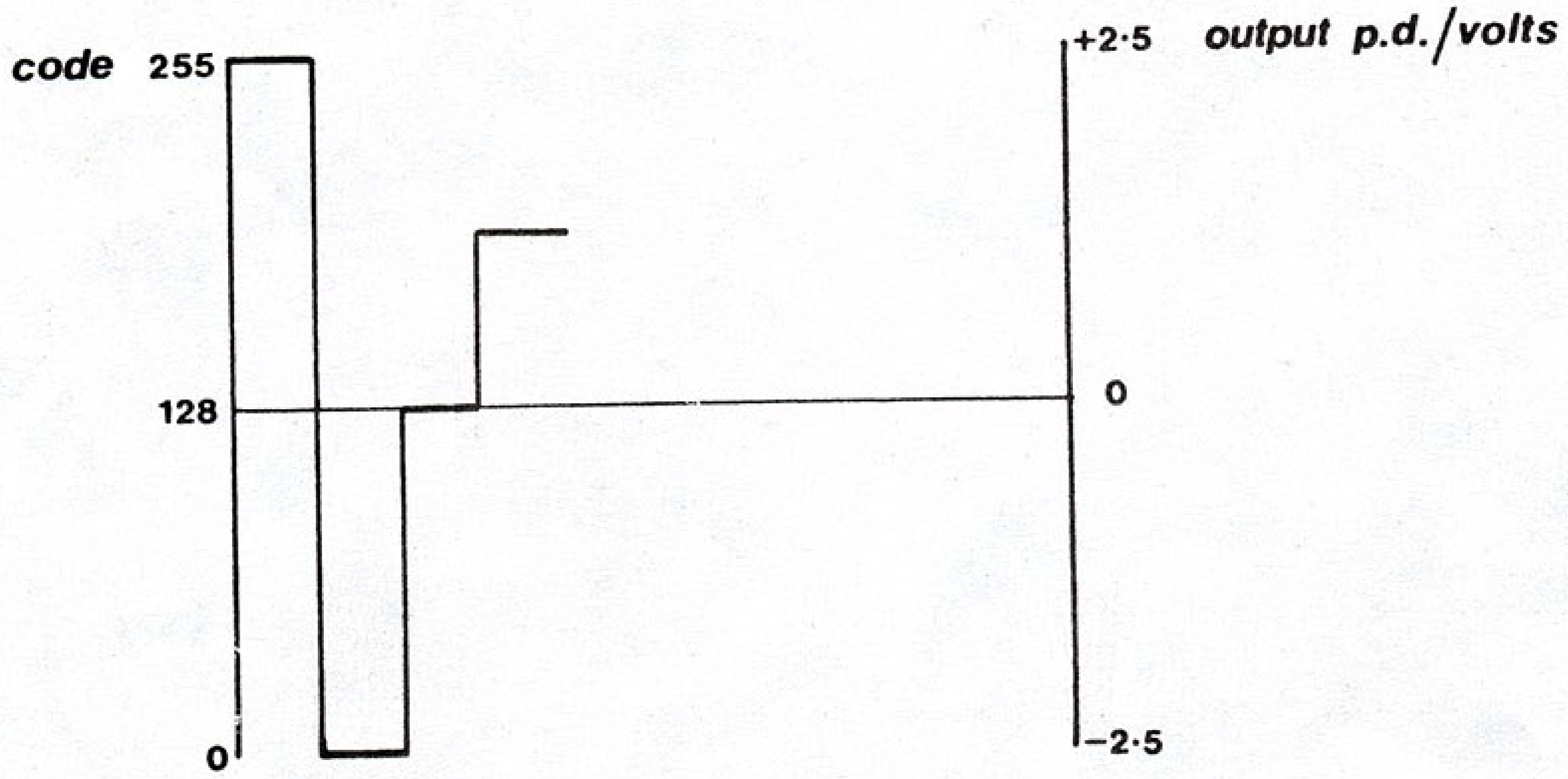
Maximum data: 256 steps.

To use this program:

1. Connect your equipment or oscilloscope to the program: output socket or digital output as appropriate;
2. Press 'RESET';
3. Press '11' to select this program;
4. Use the number pad to type in the parameter, ie the time in seconds between the output of each code (see below);
5. Press 'ENTER';
6. the display will show '1' on the left and 'xyz' on the right, meaning that the contents of memory location 1 is xyz; use the keypad to type in the code (between 0 and 255) required.
7. Press 'ENTER'; your code is now stored in memory location 1.
8. A voltage proportional to the code will appear at the analogue output, and the binary form of that number will appear on the digital output lines.

   For example, a code of 255 will give 2.5 V at the analogue output, and all the digital lines will go high, a code of 0 will give –2.5 V and all digital lines low; a code of 128 will give 0 V and the state of the digital lines will be 10000000; a code of 192 will give 1.25 V and the state of the digital lines will be 11000000, and so on. See the diagram below.

   At the same time, one or more of the three leds in the display may come on; these leds will reflect the three most significant bits of the code.



9 Press '>' to move to the next memory location, and enter the code you require in that memory location; '<' can be used to move back through memory in the same way.

10 As each code is entered, the binary form of that code is available to the digital output lines; an oscilloscope connected to the analogue output displays the waveform that has so far been built up; the oscilloscope trace will look similar to the diagram above.

11 After entering all the required codes, press 'START'; the codes will be output repetitively on both the analogue output and the digital output lines; the time for which each code is present on these output lines is determined by the parameter entered at stage 4 above.

NB It is assumed that the last code entered is the end of the sequence; if you go back to make an alteration, you must then step forwards again to the end of the sequence before pressing 'START'; the last code entered remains on the display;

12 to alter any codes,

a) press 'RESELECT DISPLAY'; the output stops;

b) press 'SCOPE';

c) press '>' or '<' as appropriate to move to relevant code;

d) alter the code as explained above;

e) after making all the required alterations, use the '>' key to step through to the final code before pressing 'START' again;

13 to change the time between the output of each code,

a) press 'RESELECT DISPLAY'; the output stops;

b) type the new time required;

c) press 'ENTER'; the output will start again automatically.

Output: At the analogue output and in binary coded form on the digital lines as described above.

USER PROGRAM CREATION – PROGRAM NUMBER 12

Description: Using this program, a user's own set of instructions can be entered into the instrument.

Parameter: None

Input: From the keyboard. Any number in the range 0 to 255.

Maximum data: 1023 codes can be entered.

Output: The user must define this in the program being written.

Brief details of how to write instructions for use with this program are in the Technical section of this manual.

INSERTION OF EXTENDED FACILITIES ROM

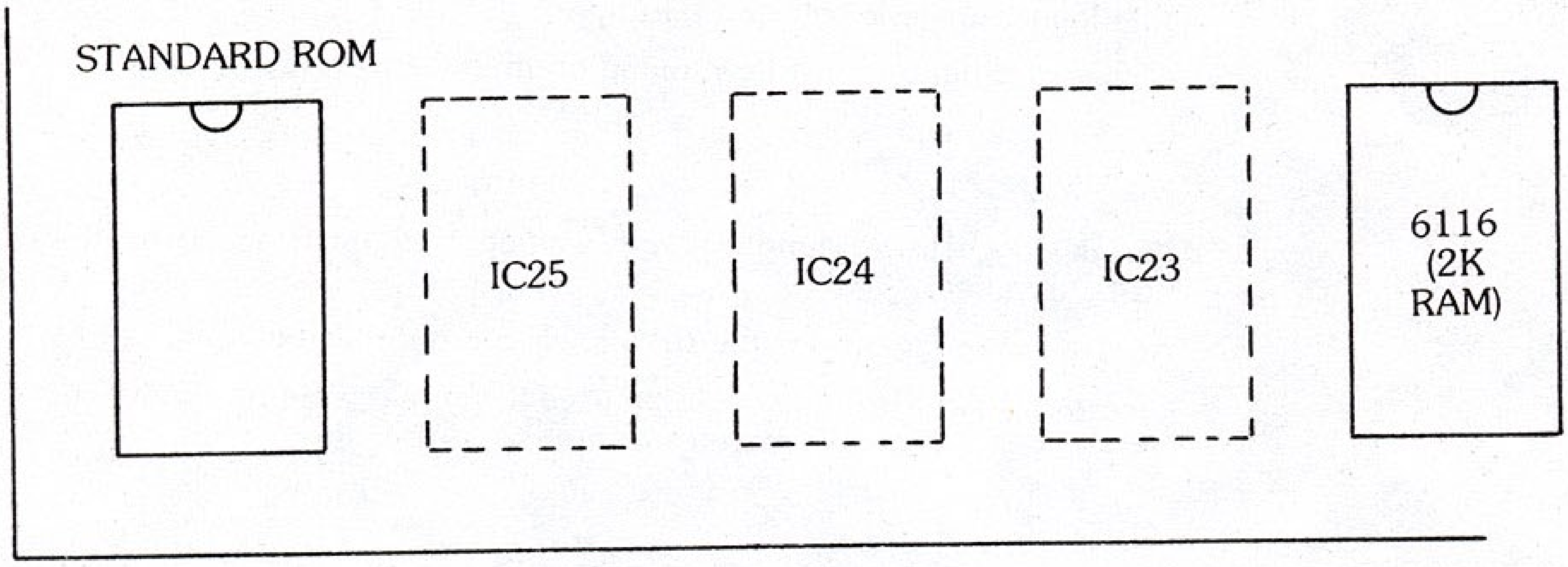
Procedure:

1 Disconnect instrument from any power supplies or other equipment.

2 Unscrew the four rubber feet supporting the logger.

3 Lift off the base plate to gain access to the printed circuit board.

4 Locate the spare EPROM socket (IC 25) into which the new EPROM is to be inserted. The position of this socket is indicated below:



5 Place the EPROM onto the socket, IC25 and check that the identation in the plastic body of the EPROM is pointing the same way as for the standard ROM.

6 Press down firmly on the printer EPROM so that the legs are held by the socket.

7 Do a quick visual check that the EPROM is in the correct way round and that all of the EPROM legs are held in the socket.

8 Refit the base plate and the feet.

9 Reconnect to power supplies and associated equipment.

FOUR CHANNEL SAMPLING CONTROLLED BY A MICROCOMPUTER AND REALTIME TRANSFER OF DATA TO MICROCOMPUTER – PROGRAM 20

Description: This program enables the logger to gather data and pass it to a host microcomputer in real time.

Parameter: Not applicable to this routine.

To use this program: When this routine is entered, it is assumed that the logger is linked to a microcomputer's parallel port by a suitable cable and that 8 data lines (into the computer) together with 2 control lines and ground are available.

A suitable cable for the BBC 'B' microcomputer is supplied with the RS data logger analysis software package.

A control program must be running on the host microcomputer.

Keypress sequence

2 0 ENTER

puts the logger into a standbye mode waiting for control signals from the host computer.

An example program for the BBC microcomputer is inc luded.

Clearly, a BASIC routine could be used if samples were required over time scales of seconds, minutes or hours.

Section 3.20

BBC MICROCOMPUTER MACHINE CODE CONTROL PROGRAM

```
 10    ?&70=0:?&71=&30:?&FE6C=&D0:?&FE62=0:?&FE6D=&FF
 50    ?&FE6E=0:HIMEM=HIMEM-&450
 90    FOR pass=0 TO 3 STEP 3
100    P%=HIMEM
110    [
120    OPT pass
130    .control
140    LDY #0
150    JSR pulse
160    .back
170    LDA #&10
180    .wait
190    BIT &FE6D
200    BEQ wait
210    LDA #&FF
220    STA &FE6D
230    LDA &FE60
240    STA (&70),Y
241    LDX #&N
242    .delay
243    DEX
244    BNE delay
245    JSR pulse
250    INY
260    BNE back
270    RTS
280    .pulse
290    LDA #&F0
300    STA &FE6C
310    LDA #&D0
320    STA &FE6C
330    RTS
340    ]
380    NEXT pass
400    CALL control
430    FOR I=1 TO 200:AD%=&2FFF+I
435    N%=?AD%
436    VL$= RIGHT$("     "+STR$(N%),4):PRINTVL$;
440    NEXT
450    END
```

NOTE:- In line 241 the user should substitute a hexadecimal number in the range 1 to FF in place of 'N'. This value determines the intersample time.

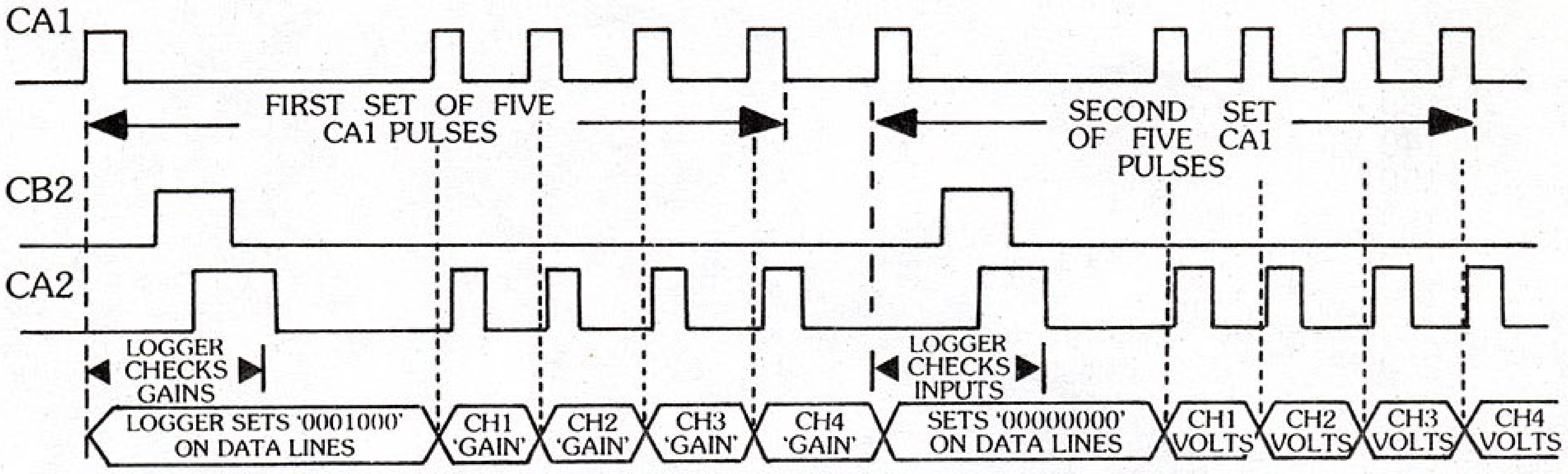
When this program is executed, 200 decimal data values will be displayed, in rows of 10 values. For example:

```
16  25  129  253  25  0  N1  N2  N3  N4

16  25  129  253  25  0  M1  M2  M3  M4 etc
```

The significance of this data format is now discussed.

TIMING OF SIGNALS ON CONTROL AND DATA LINES



When the logger receives the first positive–going pulse on CA1, it takes samples of the gains for all four analog channels and sets the data lines to '00010000' or decimal 16. When the next four pulses are sent from the controlling microcomputer, the logger places the gain values on the data lines and sends a pulse each time on CA2 to tell the microcomputer that the data on the datalines is valid.

+/–0.25 range will give values between 0 to 50
+/–2.50 range will give values between 100 to 150
+/–25.0 range will give values between 200 to 255.

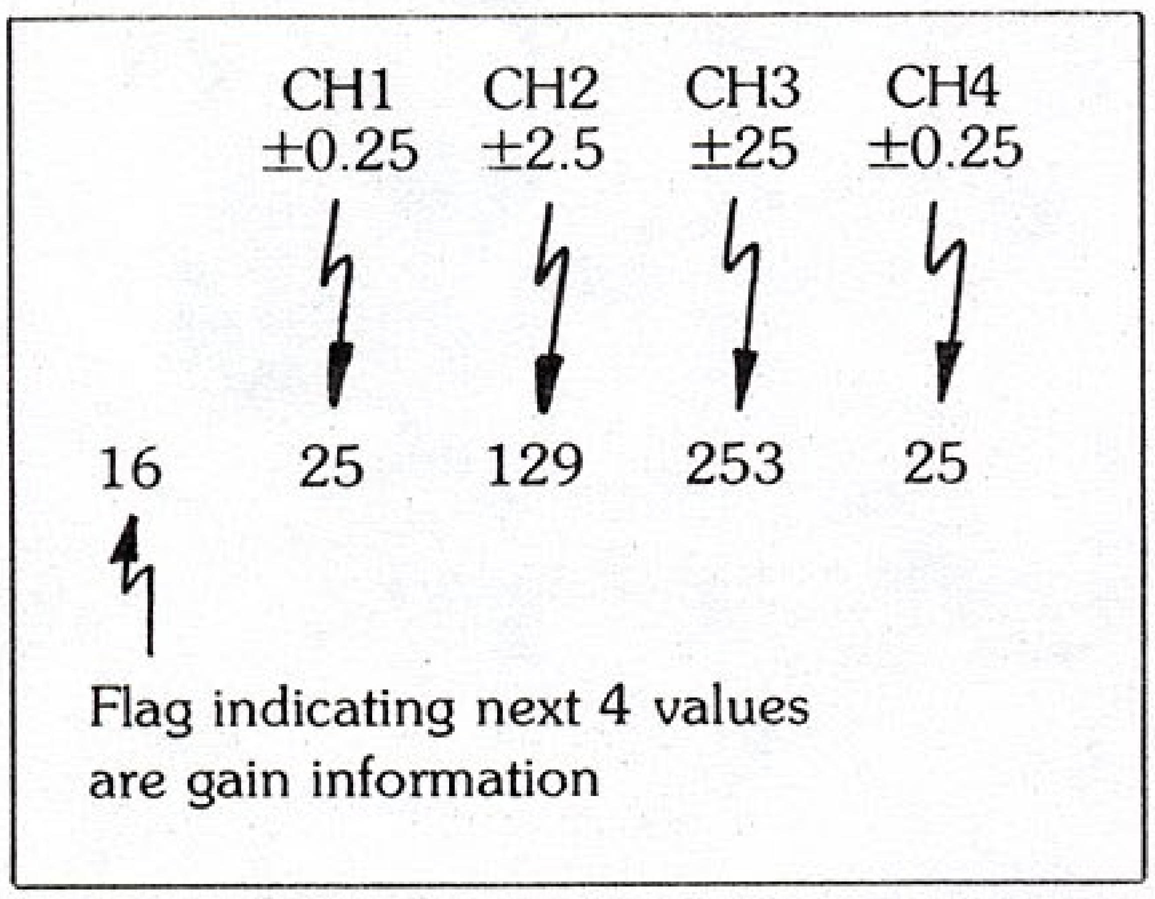
This information can then be interpreted by the host microcomputer and the gain setting of each channel ascertained.

The next five CA1 pulses initiate a rapid sampling of the actual analog voltages at the CH1, CH2, CH3 and CH4 inputs and the placing of these data bytes on the data lines.

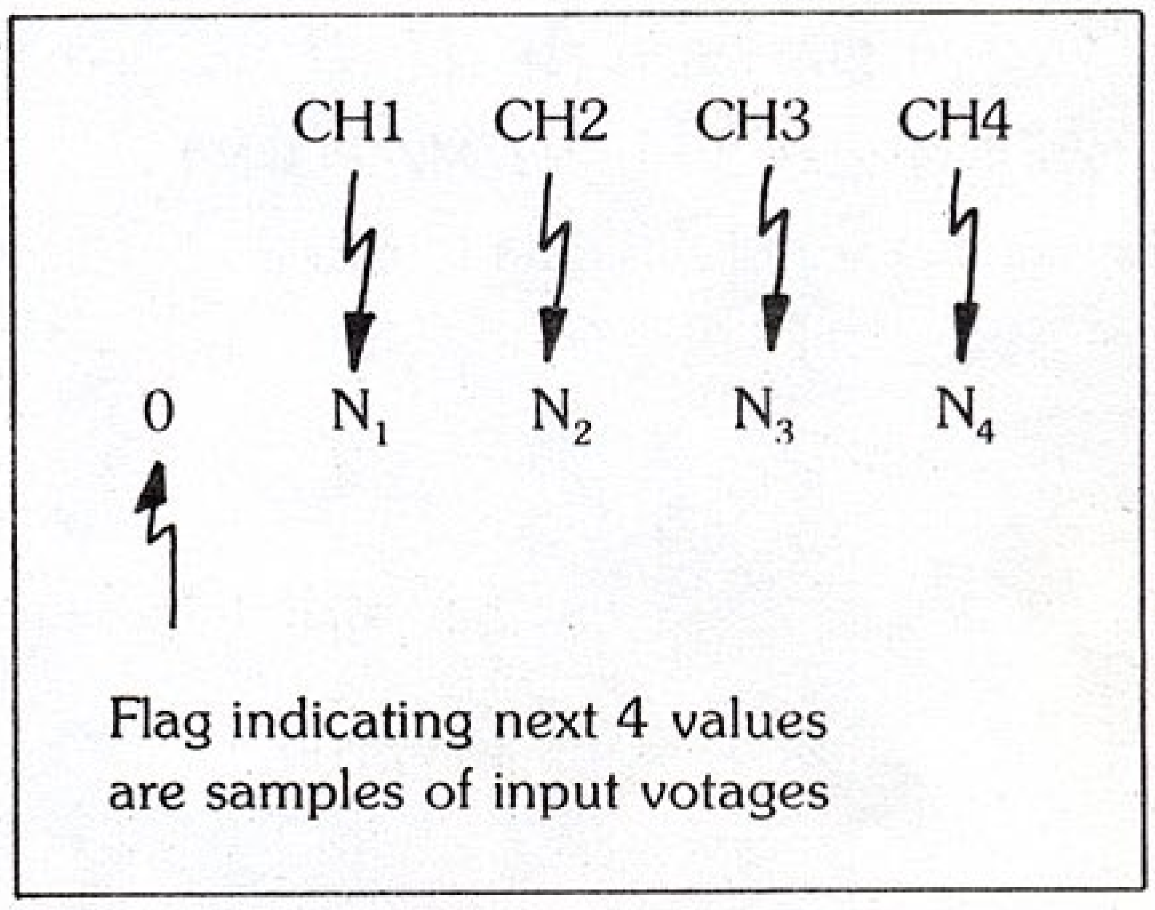
The decimal values N1, N2, N3 and N4 may be converted to the appropriate analog voltage by the relationship

$$\text{VIN} = \frac{0.25}{125}(\text{N}-128)\ \text{*GAIN}$$

$$\text{where} \begin{cases} \text{Gain} = 100 \text{ for } +/-25.0 \text{ range} \\ \text{Gain} = 10 \text{ for } +/-2.5 \text{ range} \\ \text{Gain} = 1 \text{ for } +/-0.25 \text{ range} \end{cases}$$



Flag indicating next 4 values are gain information



Flag indicating next 4 values are samples of input votages

DATA TABLE FOR SINGLE CHANNEL, FAST TRANSIENT RECORDER – PROGRAM NUMBER 21

Description: Prints a table of data (voltages) recorded in memory using program 01.

Parameter: No parameter needed if a 40 column printer is used. Select a parameter of '80' if either 80 or 120 column printers are used.

To use this program: Assuming data has already been collected by program '01', the keypress sequence:

2 1 ENTER

will produce a two line heading on the printer and requests the user to select a sample number (1 to 999) from which the datatable is to begin.

If the feature of interest is for example after the 101st sample, the required keypress sequence is now:

1 0 1 ENTER

Note that the voltage sample 101 in the 'block 2' ie "CH 2" column is actually the (1024 + 101) ie the 1125th sample of Channel 1. Similarly, the sample 101 in the 'Block 3' ie "CH 3" column is actually the (2048 + 101) ie the 2149th sample of Channel 1 and the sample 101 in the 'Block 4' column is really the (3072 + 101) ie the 3173rd sample.

During printing, the user may abort the run by pressing "STOP" for 1 second. The instrument will then flash '0-P' and the user is back into program '01' display options.

NOTE: Disconnect the printer while data logging is taking place.

DATA TABLE FOR 4-CHANNEL (MILLISECONDS) DATA LOGGER – PROGRAM NUMBER 22

Description: Prints a table of data (voltages) recorded in memory using program 02.

Parameter: No parameter is needed if a 40 column printer is used. Select '80' if either 80 or 120 column printer is used.

To use this program: Assuming data has already been collected by program '02', the keypress sequence

2 2 ENTER

will produce a two line heading on the printer and requests the user to select a sample number (1 to 999) from which the datatable is to begin. Therefore the keypress sequence:

5 4 ENTER

will print out the data from the fifty-fourth set of samples.

During printing, the user may abort the run by pressing STOP for 1 second. The printout is stopped, and the display flashes '0-P'. The user is back into program '02' display options.

NOTE: Disconnect the printer while data logging is taking place.

DATA TABLE FOR 4-CHANNEL (SECONDS) DATA LOGGER – PROGRAM NUMBER 23

Description: Prints a table of data (voltages) recorded in memory using program 03.

Parameter: No parameter is needed if a 40 column printer is used. Select a parameter of '80' if either 80 or 120 column printer is used.

To use this program: Assuming data has already been collected by program '03' the keypress sequence

2 3 ENTER

will produce a two line heading on the printer and requests the user to select a sample number (1 to 999) from which the datatable is to begin. Therefore, the keypress sequence:

8 9 ENTER

will print out the data from the eighty-ninth set of samples.

During printing, the user may abort the run by pressing "STOP" for 1 second. The printout is stopped, the display flashes '0-P' and the user is back into program '03' display options.

NOTE: Disconnect the printer while data logging is taking place.

GRAPH FOR FAST TRANSIENT DATA LOGGER – PROGRAM NUMBER 24

Description: Prints a graphical output of data (voltages) recorded in memory using program 01.

Parameter: No parameter needed if 40 column printer used – Select a parameter of '80' if either 80 or 120 column printer used.

To use this program: Assuming that data has already been collected by program '01', the keypress sequence

2 4 ENTER will produce a two line heading on the printer and requests the user to select the sample number (1 to 999) from which the printout is to begin. Press ENTER to continue printing.

During printing, the user may abort the run by pressing STOP for 1 second. The printout is stopped, the display flashes 'O-P' and the user is back into the '01' display options.

Note that to output 1K of samples requires a considerable length of paper. If compact graphs are required it is usual to considerably increase the intersample time compared with that used for oscilloscope display.

NOTE: Disconnect the printer while data logging is taking place.

GRAPH FOR 4 CHANNEL (MILLISECONDS) DATA LOGGER – PROGRAM NUMBER 25

Description: Prints a graphical output of data (voltages) recorded in memory using program 02.

Parameter: No parameter needed if 40 column printer used. Select a parameter of '80' if either 80 or 120 column printer used.

To use this program: Assuming that data has already been collected by program '02', the keypress sequence

2 5 ENTER

will produce a two line heading on the printer and requests the user to press the required channel number.

When another four lines have been printed, select CH1, CH2, CH3 or CH4 and the printer will respond with another request – this time for the required sample number (1 to 999) from which the graphical printout is to begin. Press ENTER to continue printing.

During printing, the user may abort the run by pressing STOP for 1 second. The printout is stopped, the display flashes '0-P' and the user is back into the '02' display options.

Note that to output 1K of samples requires a considerable length of paper. If compact graphs are required it is usual to considerably increase the intersample time compared with that used for oscilloscope display.

NOTE: Disconnect printer while data logging is taking place.

GRAPH FOR 4 CHANNEL (SECONDS) DATA LOGGER – PROGRAM NUMBER 26

Description: Prints a graphical output of data (voltages) recorded in memory using program 03.

Parameter: No parameter needed if a 40 column printer is used. Select a parameter of '80' if either 80 or 120 column printer is used.

To use this program: Assuming that data has already been collected by program '03', the keypress sequence

2 6 ENTER

will produce a two line heading on the printer and requests the user to press the required channel number.

When another four lines have been printed, select CH1, CH2, CH3 or CH4 and the printer should respond with another request – this time for the required sample number (1 to 999) from which the graphical output is to begin. Press ENTER to continue printing.

During printing, the user may abort the run by pressing STOP for 1 second. The printout is stopped, the display flashes '0-P' and the user is back into the program '03' display options.

Note that to output 1K of samples requires a considerable length of paper. If compact graphs are required, it is usual to considerably increase the intersample time compared with that used for oscilloscope display.

NOTE: Disconnect printer while data logging is taking place.

FOUR CHANNEL (MINUTES) DATA LOGGER WITH DIRECT OUTPUT TO PRINTER – PROGRAM NUMBER 27

Description: This program reads voltages on all 4 channels and records them directly on a suitable printer at user defined intervals.

Parameter: Any value from 0 to 999. If parameter '0' is selected, the voltages at each of the four analogue inputs is sampled every 5 seconds. Any other parameter 'n' gives a line of printout every 'n MINUTES'.

To use this program: Press

2 7 n ENTER

The printout should give a suitable heading showing the sampling interval.

The datalogging begins by pressing START. To stop logging data, press STOP for 1 second. The user may then select all of the output options as for the standard datalogging routines on the standard ROM.

The datalogging will automatically end when 1018 bytes of information for each analogue channel have been stored.

PRINTOUT DATA TABLE FOR THE MULTICHANNEL TIMER PROGRAM 06 – PROGRAM NUMBER 28

Parameter: No parameter is needed if a 40 column printeris used. Select a parameter of '80' if either 80 or 120 column printers are used.

To use this program: The multichannel timer program '06' times any changes of eight data lines (pins 15 to 21 and 23) going into the digital input/output socket.

After the data has been collected using program '06', press

RESET and then select this printout program

2 8 ENTER

The printing will stop when all valid code changes and the associated times have been printed. The logger will now flash '0-P' and the user will be able to call up all of the output options for program '06' once more.

NOTE: Do not have the printer connected when running program 06 as results will be meaningless.

4.1 Overview of the System

A full wave bridge rectifier and regulator Integrated circuit ensure that the stabilised +5 volts is obtained from either AC (8 volts – 9 volts) or DC (8 volts–14 volts). It is not advisable to exceed the upper limit because the current drawn is between 0.45 and 0.6 amps (depending upon the number of 7 segment displays activated) and the power dissipated in the regulator will become excessive. The design therefore allows the logger to be operated by 4AH Nicad's on field trips or a low voltage supply in the laboratory. A mains adaptor is supplied with the instrument. Note that, if powered by a low voltage dc supply, the polarity of the power leads is unimportant.

The circuit board is connected to the touch sensitive keypad via a flexible multitrack strip.

The Logger is based on the 6802 central processor unit which has an on chip oscillator and 127 dec RAM locations. The software is held in eraseable, programmable, read only memories (EPROMs) which means that the software defining all of the functions is up and running when power is supplied. The initial software contains 4096 8 bit codes. The unit has 4096 RAM memory locations provided by the two 6116 integrated circuits, and it interacts with the outside world via 3 PIAs. The memory map is shown in figure 5. Each PIA contains six registers and their hexadecimal addresses are shown in Table 2.

HEXADECIMAL ADDRESS	REGISTER NAMES
$X000	Output Register A (and Data Direction Register A)
$X001	Output Register B (and Data Direction Register B)
$X002	Control Register A (b_2 = 0 selects DDRA, b_2 = 1 selects 0RA)
$X003	Control Register B (b_2 = 0 selects DDRB, b_2 = 1 selects ORB)

'X' stands for C, D and E

TABLE 2

A more complete description of the function of the PIA data lines and the control lines is given in Table 4. When power is applied, the CPU initialises the PIA's, defines the stack at the top of the CPU RAM space, ie the Stack Pointer is $007B, displays 'HELLO' for a few seconds and then the 'program request' prompt "P-" is displayed. If the user requests a program number between 00 and 16 dec, the CPU picks up the vector address of the appropriate routine from a pair of consecutive memory locations between $FA00 and $FA22. (If the user accidentally requests a program number outside this range, the unit will react in an unpredictable way – until further EPROM's have been inserted, see section 2.3. In order to regain control, either press RESET or, if this does not give "P-" on the display, switch the power off and start again.)

PIA HEXADECIMAL ADDRESS	DATA OR CONTROL	FUNCTION
$C000	8 DATA	Inputs from keypad and 74C922
$C001	8 DATA	Outputs to display driver
SC002	CA1	Digital control input
	CA2	DP output to display driver
$C003	CB1	Data available pulse input from 74C922
	CB2	Write pulse to display driver
$D000	8 DATA	Outputs to DAC
$D001	8 DATA	Inputs from ADC
$D002	CA1	Pulse input
	CA2	Sync output
$D003	CB1	End of conversion (from ADC)
	CB2	Start Conversion (to ADC)
$E000	8 DATA	Digital inputs
$E001	8 DATA	Digital outputs (& LED drivers)
$E002	CA1	Digital control input
	CA2	Select analogue channel
$E003	CB1	Digital control input
	CB2	Select analogue channel

TABLE 4

Each of the four analogue inputs and the pulse input has an input impedance of 1 Mohm in order to minimize the loading on external sensors. Each of these inputs is protected against input voltages of up to +/–250 volts, and the three switched gain settings give effectively, a dynamic range of +/–250 millivolts, +/–2.5 volts and +/–25 volts. The software senses the manual switch positions and automatically adjusts the displayed decimal point for the digital voltmeter (program '00') and transient recorder programs (01, 02, 03). Note, however, that the maximum voltage swing at the 4 mm "scope output" socket is +/–2.5 volts and therefore, if the middle gain range has been selected during data logging programs, the voltages replayed to the oscilloscope will be facsimilies of the input waveform.

Because of the relatively high input impedance of the analogue and pulse channels, it is possible that there may be crosstalk or interaction between the pulse channel and neighbouring analogue channels. Therefore, for best results, the user should avoid inputting pulses to 'PULSE INPUT' during the data capture phase of transient recorder programs. When the channel gain is switched to give +/–250 millivolts dynamic range, the digitization step of the analogue to digital converter corresponds to 2 millivolts. The output from some sensors, eg thermocouples, will require a stage of voltage amplification before the signal can be entered into the Logger.

In the data logging programs (02 and 03) the analogue channels are sequentially selected in the order 1, 2, 3 and 4 by the multiplexer and the voltage is transferred to the ZN448 ADC.

The ADC's clock runs at 1 MHz, and therefore the digitization process takes approximately 9 microseconds. However, in order to pick up the 8 bit code and store it in the next memory location and check for the end of memory, the shortest period between two consecutive samples is 26 microseconds. This is the intersampling time when the unit is data logging with program '01' and parameter '0' and give a sampling frequency of 38 Khz. If analogue signals with significant frequency components above 19 Khz are attempted to be logged at this rate aliassing will result.

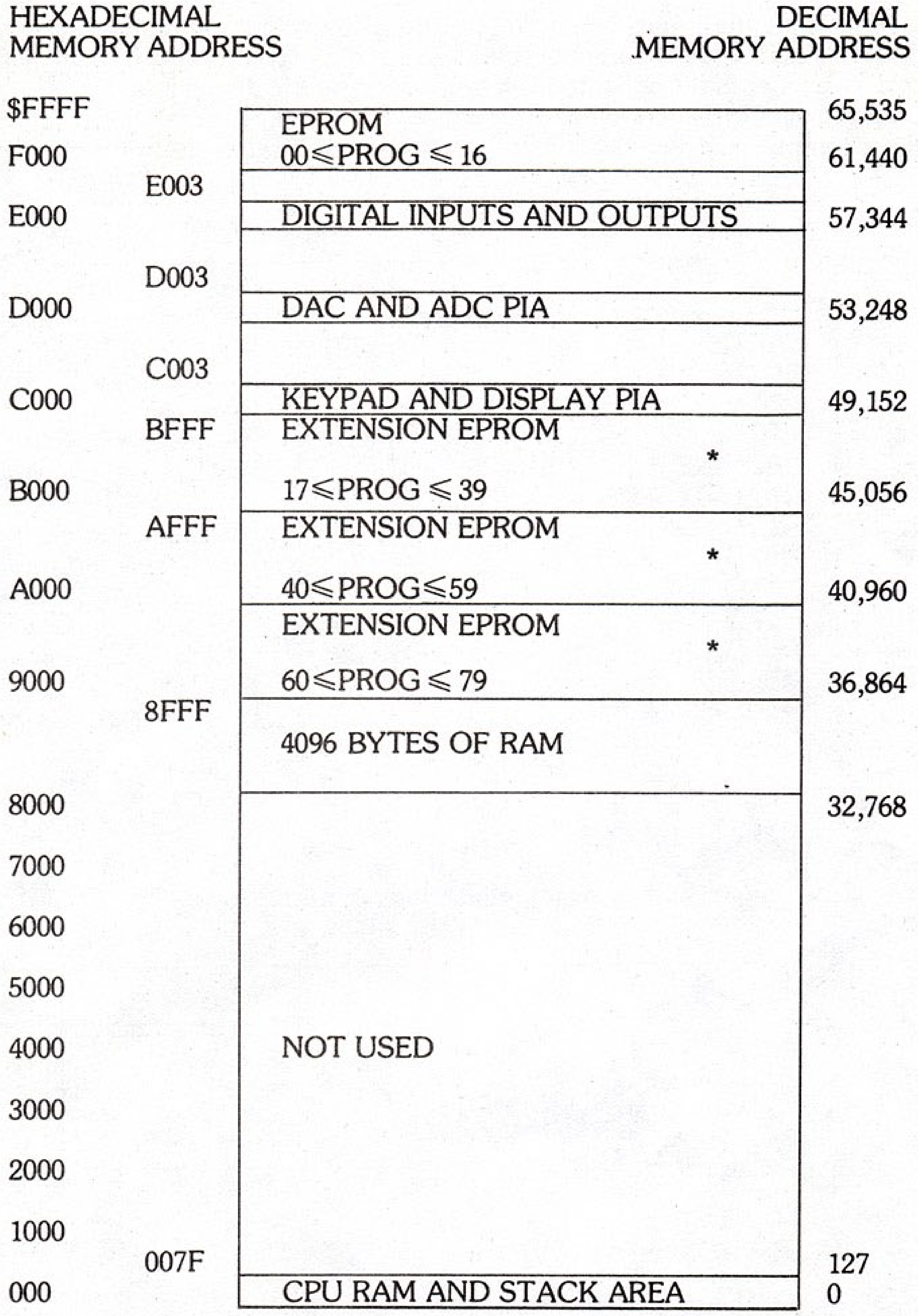


FIGURE 5: MEMORY MAP

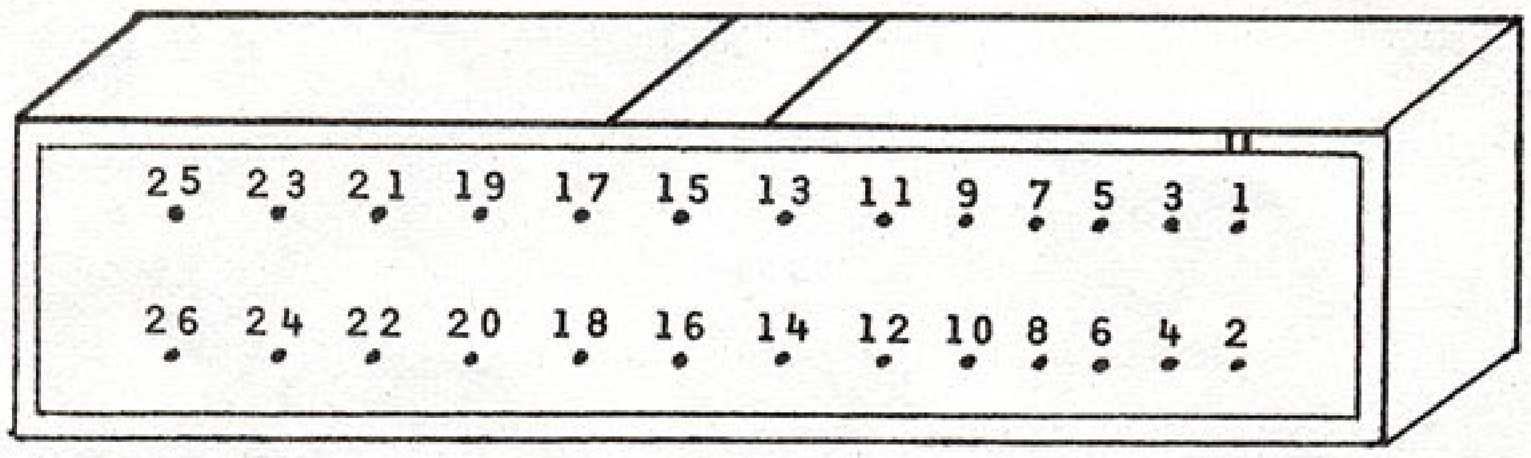
* A 6116 RAM chip could be used in place of any EPROM. (The RAM MEMORY defined being $9800-$9FFF, $A800-$AFFF or $B800-$BFFF)

Although the analogue inputs are buffered, the digital input/output port is not buffered. These inputs and outputs are TTL compatible and as such could interface directly with microcomputer 'user ports' or printer input ports. If it is intended to use the digital lines to drive relays or lamps or motors, a power driver stage will be required, (eg Darlington drivers, stock number 307–092). The digital input/output port pin description and pin identification is shown in Table 5 and Figure 6.

PIN	DESCRIPTION	PIN	DESCRIPTION
1	EARTH	14	PB0 DATA In/Out
2	EARTH	15	PA7 DATA In/Out
3	+5 volts	16	PA6 DATA In/Out
4	EARTH	17	PA5 DATA In/Out
5	CB2 ($E003) Control In/Out	18	PA4 DATA In/Out
6	CB1 ($E003) Control In/Out		19 PA3 DATA In/Out
7	PB7 DATA In/Out	20	PA2 DATA In/Out
8	PB6 DATA In/Out	21	PA1 DATA In/Out
9	PB5 DATA In/Out	22	CA2 ($E002) Control In/Out
10	PB4 DATA In/Out	23	PA0 DATA In/Out
11	PB3 DATA In/Out	24	CA1 ($E002) Control Input
12	PB2 DATA In/Out	25	EARTH
13	PB1 DATA In/Out	26	CA1 ($C002) Control Input

TABLE 5

Note Pin 1 of the plug is marked by a triangle, the polarising socket being towards the base of the unit.



VIEW OF SOCKET FROM SIDE

FIGURE 6: DIGITAL INPUT/OUTPUT PORT

Although the PIA data lines shown in Table 5 could be programmed as either inputs or outputs, the convention adopted in the programs in the standard EPROM is to assign PBO–PB7 as output lines and PAO–PA7 as input lines. Therefore the voltage measured on line PBO corresponds to the status of the least significant bit of the code stored in memory location $E001 and the voltage on line PB7 corresponds to the status of the most significant bit of the code stored in $E001. Similarly, the external voltage (+5 volts or 0 volts) applied to the line PAO will determine the status of the least significant bit of the code in $E000 and the external voltage applied to the line PA7 will determine the status of the most significant bit in $E000.

The eight digit 7 segment displays are controlled by a Universal LED Driver integrated circuit. Included in this device is an 8 x 8 static memory array providing storage for the displayed information and all of the multiplex scan circuitry (to minimize the power drain) and the high power digit and segment drivers. The display driver is controlled by the PIA at $C000. Most of the keys are scanned by the 74C922 keypad encoder but four of the keys when pressed define a low voltage on one of four PIA data lines ($C000). The four keys in question are

START   STOP   MICRO   and   <<>>

4.2 Software Expansion

Your only reason for opening up the instrument should be to extend the on-board software, as it becomes available, by inserting extra 2732 EPROM's into the sockets provided. For instance the RS Extended Facilities ROM stock number 611–981 can be used to extend the instrument's facilities. CARE must be exercised when disengaging the box top from the base, and the following procedure is recommended:

i) Make sure that the power lead is disconnected.

ii) Remove the screws on the base of the unit.

iii) On removing the base you will see the row of sockets next to the EPROM labelled RS Stock number 611–975.

iv) The EPROM must be a type 2732 and must be inserted in the correct socket, the correct way round – as shown in figure 7.

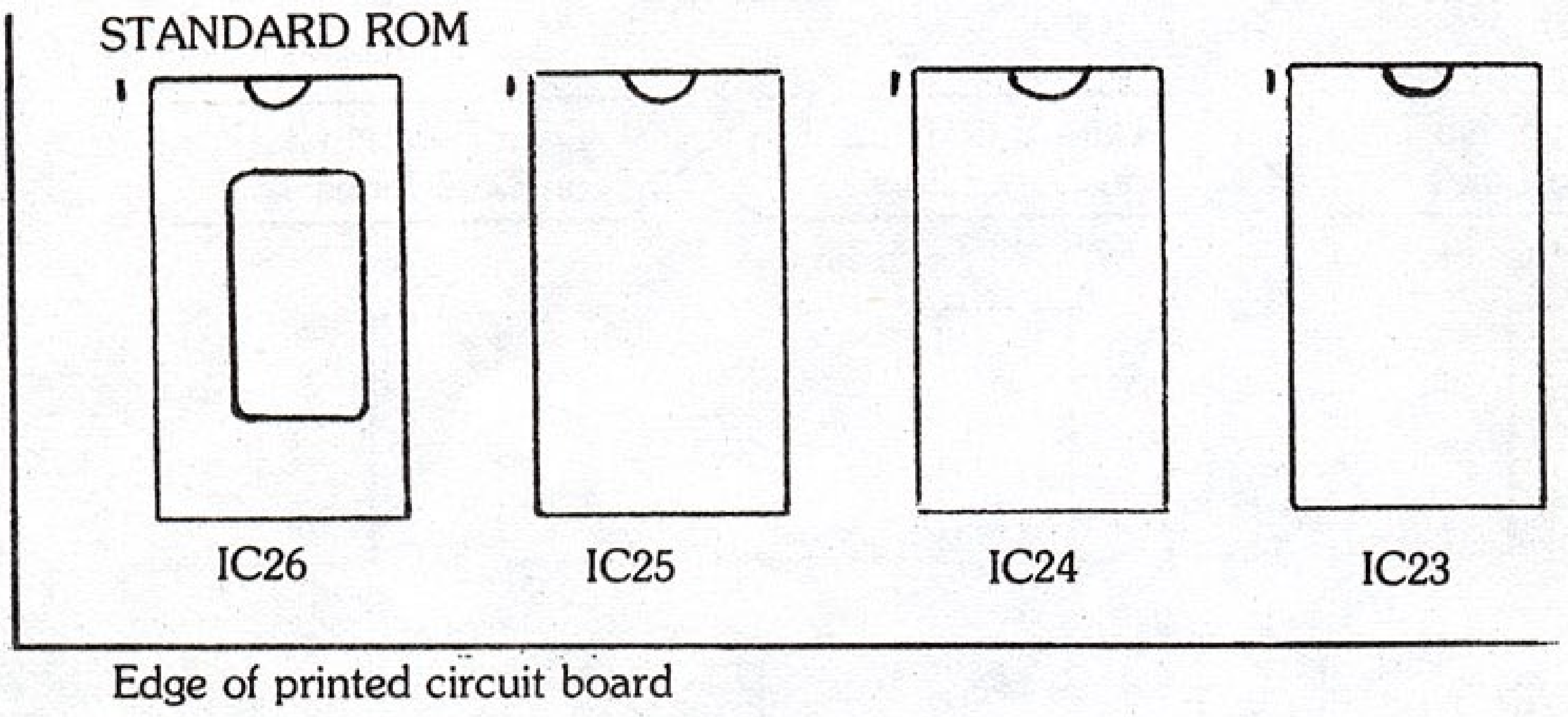


FIGURE 7: EPROM ORIENTATION

v) Do a quick visual check to ensure that all of the EPROM pins are seated in the socket holes and press down on the EPROM to make sure that it is held firmly by the socket.

vi) Reassemble the instrument.

User Created Eproms:-

It was always intended that a user, having tested a program in RAM (as described later) and having EPROM creation facilities, should be able to insert his own EPROM into socket IC24. The CPU expects to find the vector address of the start of the user routine at specific locations within the EPROM memory space. The range of locations assigned to the vector addresses is $AF50 – $AF77 inclusive. Let's take a specific example: if the user wants to start his routine at the lowest EPROM address $A000 and to call up this program with the two digit number 40, the user MUST place the most significant byte of the vector address ($A0) in memory location $AF50 and the least significant byte of the vector address ($00) in memory location $AF51. Similarly, if the user designates a routine starting at $A123 as program '41', the user MUST place $A1 in memory location $AF52 and $23 in memory location $AF53.

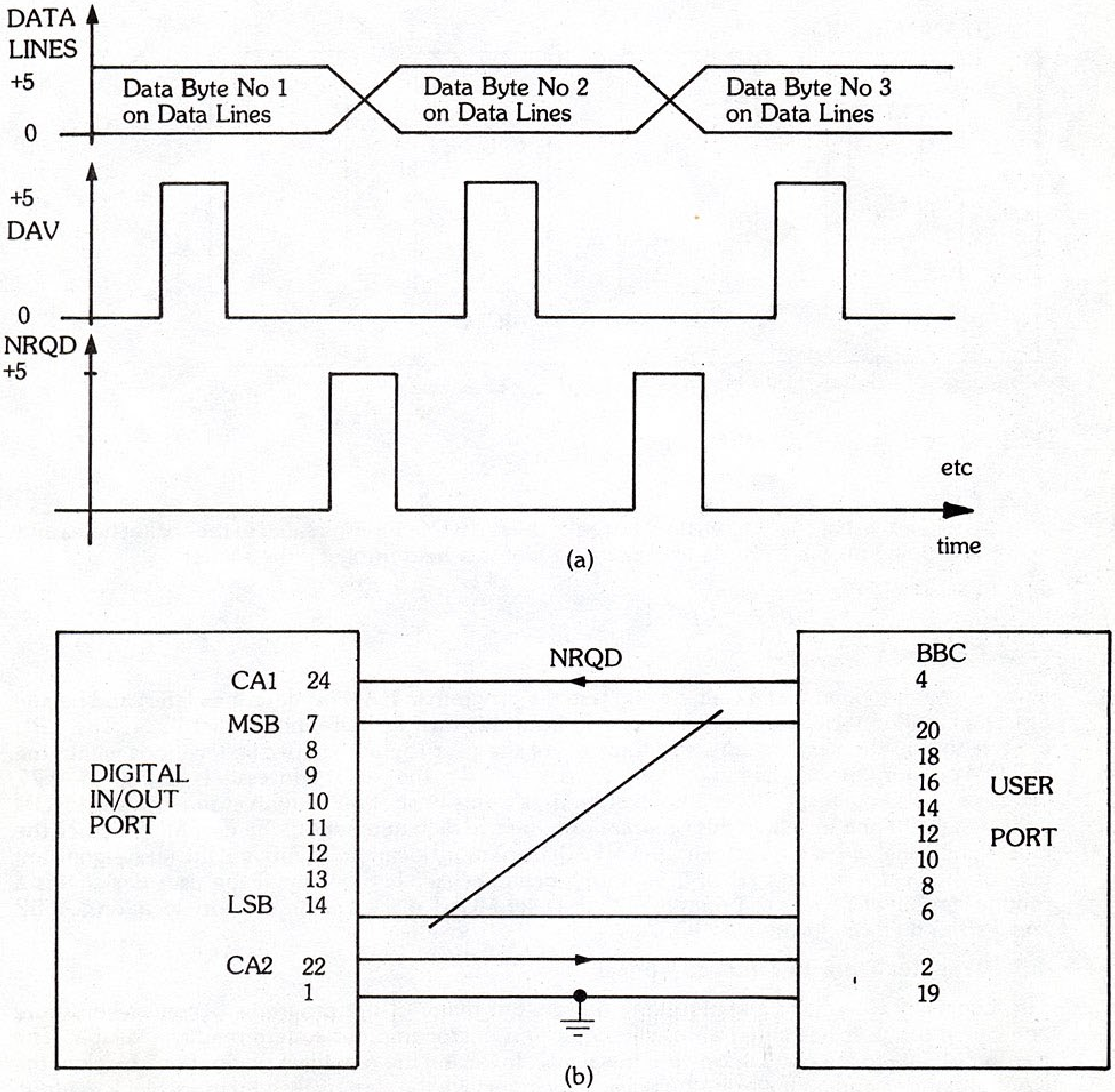
4.3 Transfer Data to Microcomputer

The Logger is essentially a stand alone device, but many of the programs become even more effective if the user has either an oscilloscope or a microcomputer system readily available. The transfer of data to the oscilloscope is a trivial task, involving the repetitive readout of a block of the RAM's memory, and a synchronising pulse coincident with the start of each memory block readout, to facilitate a steady oscilloscope trace.

Section 4.3

Data is transferred between microcomputer and peripheral devices either via a serial link or a parallel link. The technique adopted here is to use a parallel link where each bit of an 8 bit code defines the voltage on one of 8 data lines and the sender keeps in synchronism with the receiver (a microcomputer) by means of two control lines. One of the control lines is energised by the sender just after a valid 8 bit code has been placed on the data lines. This pulse from the sender alerts the receiver to the fact that the correct code is on the data lines. The receiver then reads the data, stores it and energises the other control line with a positive voltage pulse. When the sender detects this pulse, it knows that the previous data code has been picked up and it can now replace the previous data code by the next valid code. The cycle is then repeated, as shown in figures 8(a). The transfer of data in this way is called a 'Handshake', and the receiver must have a special linker routine at the start of its data processing program in order to synchronise its operation with the sender.

For fast machine code data transfer, plotting and analysis software for the BBC microcomputer see RS Data Logger analysis software, Stock number 611–997.



(a)

(b)

FIGURE 8: HANDSHAKING DATA BETWEEN LOGGER AND MICROCOMPUTERS

In theory, any microcomputer with eight data lines which can be defined as inputs and one (but preferably two) control lines can be linked to the Logger.

Section 4.3

Handshake Protocol

When a block of data is transferred to a microcomputer, the 'receiving' microcomputer must not only have the simple 'linker' routine to coordinate the transfer, but there must be an agreed protocol within the data bytes. The size of the block of data depends upon the program selected, therefore, the logger must somehow tell the receiving microcomputer how many data bytes are to be transferred on the parallel link. Also, in order to file and process the data received, the microcomputer must know:

i) which logger program generated the data

ii) which parameter was chosen

iii) which channel or block was selected for readout (1, 2, 3 or 4)

iv) the gain setting of the manual switch (when appropriate).

The protocol adopted for the data transfer is therefore

NH	NL	PR	PARH	PARL	BLOCK	GAIN	FIRST DATA BYTE	SECOND DATA BYTE

time →

time where NH is the value of the first byte transferred, NL is the second byte transferred, etc.

The number of data bytes in the record is (256NH + NL) which must be differentiated from the total number of bytes transferred ie (256NH + NL + 7). The third byte transferred is PR and this represents the program number selected. The fourth and fifth bytes transferred, PARH and PARL are the high and low byte of the parameter selected. Therefore the parameter value is given by (256PARH + PARL). The sixth byte called BLOCK contains 1, 2, 3 or 4 and represents either the analogue channel whose data values are to follow or a particular block of data (see 3.3). The seventh byte is necessary to define the gain setting during the TRANSIENT RECORDER programs – see below.

Data Formats

The structure of the data transferred depends upon the program number selected.

1) TRANSIENT RECORDER (01, 02 AND 03)

After the 7 data byte preamble, the data bytes are outputted sequentially and in blocks of 1023dec bytes.

Note that the data transferred is in the form of an 8 bit code which defines a certain voltage value sensed by the ADC. The seventh byte transferred notifies the receiving microcomputer of the channel gain, G, during datalogging, and is either set 1, 10dec or 100dec, depending upon the chosen dynamic range of +/–0.25 volts, ;/–2.5 volts or +/–25.0 volts. The conversion from transferred data value to volts seen at the input is therefore given by

volts 0.25*(datadec – 128)* G/125

ii) MULTICHANNEL TIMER (06)

After the 7 byte preamble, the data bytes are organised in the following way:

FIRST CODE SEEN	0	0	SECOND CODE SEEN	T_H	T_L

time →

The data bytes come in sets of three bytes; the first byte is the 8 bit code corresponding to the voltages on each of the 8 input data lines (at $E000) and the next two bytes contain TH, the most significant byte and TL, the least significant byte of the time in milliseconds, Tm when the previous 8 bit code had been detected.

T = 256 T + T milliseconds. m H L

USER PROGRAM CREATION (Program 12)

The most elementary type of program is a sequence of 8 bit codes. This is called a machine code program. The decimal equivalent of each 8 bit code is a number between 0 and 255dec. As the Logger can only accept decimal data via the keypad see note (1) the User Program is composed of a set of decimal numbers in consecutive memory locations. The first program instruction code MUST be placed at the displayed memory address '1' and the maximum number of program codes is 1023dec.

In order to create one's own program, program number '12' must be entered (no parameter is necessary at this stage)(2). The display goes momentarily blank and then the memory location 1 appears in the centre of the display and the contents of that location appears on the right hand side of the display. If the code in the memory location is the correct one, press '>' to move onto the next location. If a new code is required at this memory location, simply type in the new code and press ENTER. The display momentarily flickers when the new code replaces the old code in that memory location. (If you make a mistake while typing the code, press ENTER and then retype in the correct code and ENTER again.) The user can now press '>' to move onto the next memory location of '<' to check the previous memory location's contents. In this way, the sequence of decimal equivalent codes can be defined.

Note that if a decimal code greater than 255 is entered, a '1' will be placed in that memory location. The user program will be executed as soon as the START button is depressed, and if for some reason you want to stop your program, the only way is to press RESET. Your program may then be altered or checked out using the TRACE facility (see section 4.2), requesting program 16 again and pressing ENTER.

An example of a nontrivial program which is easily created by the user is shown in figure 10. This program generates a triangular waveform whose frequency is approximately 55 Hz. The program uses one of the subroutines in the on board EPROM in order to output an analogue voltage to the oscilloscope. The addresses of other useful routines are shown in table 8.

The CPU inside the instrument is the 6802 and there are a number of special registers within the CPU which do not have an assigned memory location. These registers are shown in figure 9.

Note (1) There exists a routine in an additional EPROM which could allow a user creation program to be entered in hexadecimal codes (and which displays the hexadecimal addresses).

Note (2) If program number '16' is followed by a parameter n n n, it is assumed that the user wishes to jump to memory location n n n of the machine code program in order to verify (and possibly change) the code at that location.

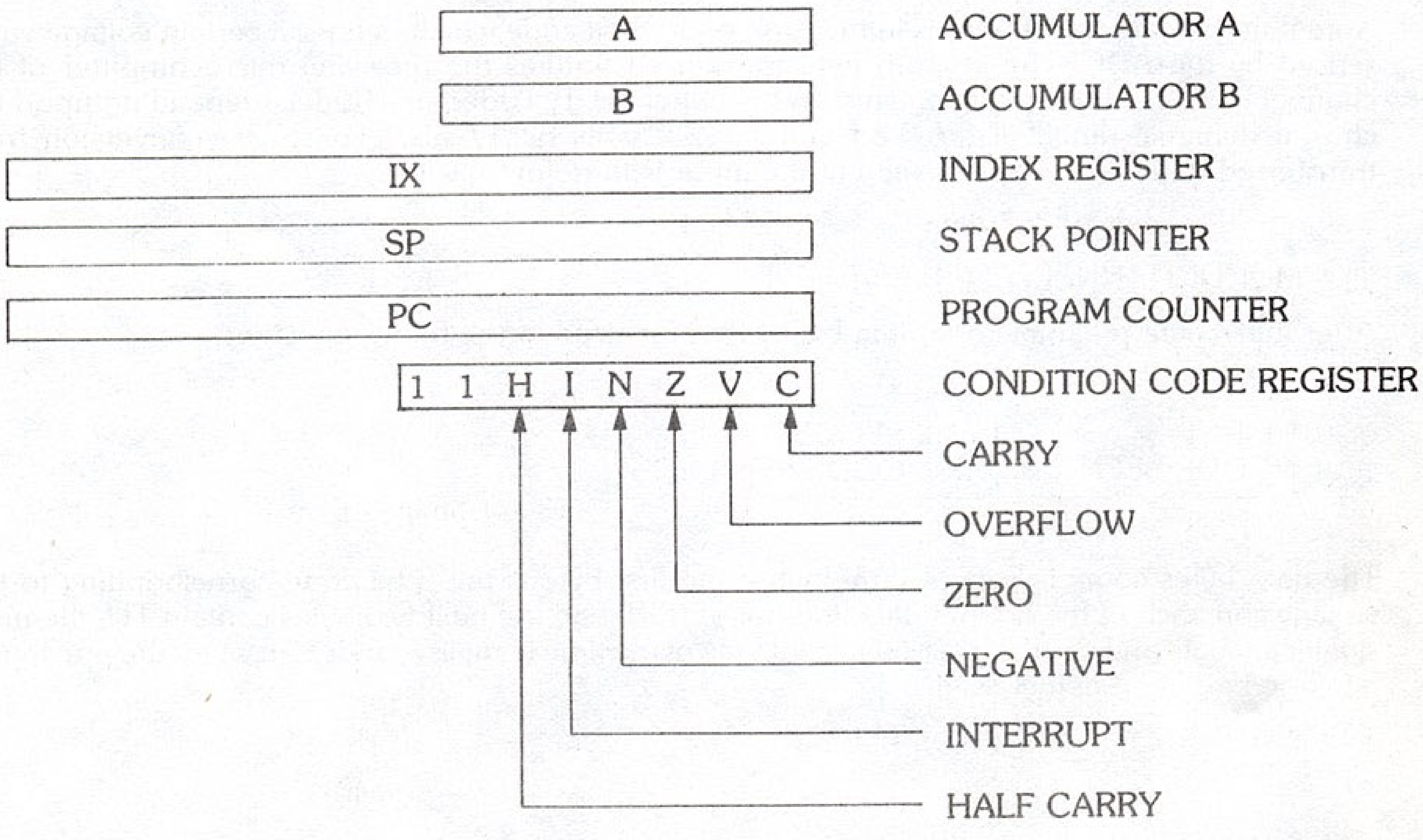


FIGURE 9: 6802 CPU REGISTERS

Most of the arithmetical and logical operations are performed in the 8 bit registers A and B. Therefore, the CPU has to fetch data from the addressable memory locations and place the data into these special registers so that the data can be processed and then returned to external memory locations. Although there are a relatively small number of distinct operations that the CPU can perform, there are many ways of acquiring data. The different ways of acquiring data are called addressing modes.

For example, the operation code for "load data into accumulator A" is either 134 or 150 or 166 or 182. If the code 134dec is followed by the code 18dec, this instructs the CPU to load accumulator A with the data value 18dec. If however, the code 150dec is followed by 18dec, this would instruct the CPU to load accumulator A with the data in memory location $0012 (which is the 18th address in the micro's memory space.) If the code 166dec is followed by 18dec, the CPU would fetch data from a memory location whose address was the eighteenth after the address specified by the contents of the index register. Finally, if the code 182dec were used, it would have to be followed by two codes and these two codes would specify the memory address from which to fetch data. (Note that in machine code, the most siginificant byte preceeds the least significant byte.)

The program counter is a 16 bit register which keeps track of the memory address of the next executable instruction in the program which is being run.

The stack point is a 16 bit register which keeps track of the location in the stack area of memory where data can be temporarily stored.

The index register is a 16 bit register which can be used either as a countup or countdown register.

The condition code register is an 8 bit register whose two most significant bits are always '1' and whose remaining six bits are independent flags which are set or cleared depending on the instruction being performed. There are many branch instructions (see figure 14) which can be used to alter the program counter (and hence the program flow) on the basis of one or more of these flags being set. The programmer may want the CPU to branch forwards OR backwards. The convention followed by Motorola is that if the most significant bit of the code following the branch opcode is '0', this code will represent a branch FORWARD. Therefore the maximum number of steps foward is 127dec. If the most significant bit of the code following the branch opcode is '1', this code will represent a branch BACKWARD. In table 9, the decimal codes required for both forward (+ve) and backward (–ve) branches are tabulated, eg if you want to BRANCH ALWAYS BACKWARDS BY 35 STEPS, look up the code for BRANCH ALWAYS, ie 32dec and the code for –35, ie 221dec and therefore the coded instruction becomes 32,221. Another reason for table 9 is that assembly language programmers are used to hexadecimal codes and for certain instructions, a ready reckoner from hexadecimal to decimal is desirble. For example, if the JUMP instruction is used, it must be followed by the complete address where the CPU is to jump to. If we wanted JUMP TO ADDRESS $8157, we would find the opcode for JUMP ie 126dec and the memory address would have to be split into the most significant byte $81 (129dec) and the least significant byte $57 (87dec). The instruction would therefore be coded as JUMP TO $8157 126, 129, 87.

The reader should refer to a Assembly lanuage Programming Manual for a complete description of the operation codes. Stock number 903–397.

The output waveform generated by the program should have the appearance of figure 11 at the analogue output socket.

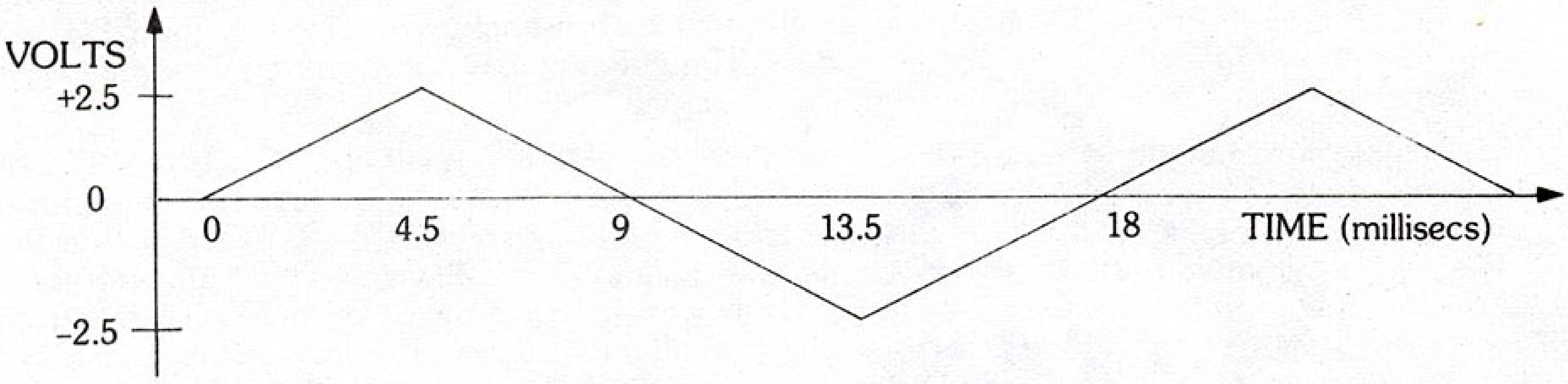


FIGURE 11: TRIANGULAR WAVE GENERATION

FIGURE 11: TRIANGULAR WAVE GENERATION

Trace Facility

During program development, it is essential to have the ability to halt the program at a certain point and to then interrogate the CPU registers in order to see if they have their expected values. An elementary 'trace' facility such as this has been provided and it is entered whenever the CPU meets a software interrupt (SWI) code as the next executable instruction within the user's program. The decimal equivalent SWI code is 63dec.

When the CPU detects this code, it stores its registers in the 'stack', blanks the display and then displays the decimal value of the program counter (PC) when the SWI code was seen. Successive FWD keypresses will display the contents of the other CPU registers in the order shown in table 7.

DISPLAY	RANGE OF VALUES
PROGRAM COUNTER	35,841→36,863
INDEX REGISTER	0 → 65,535
ACCUMULATOR A	0 → 255
ACCUMULATOR B	0 → 255
CONDITION CODE REGISTER	11XXXXXX*
STACK POINTER	80 → 123
PROGRAM COUNTER	etc

* X DENOTES 1 OR 0

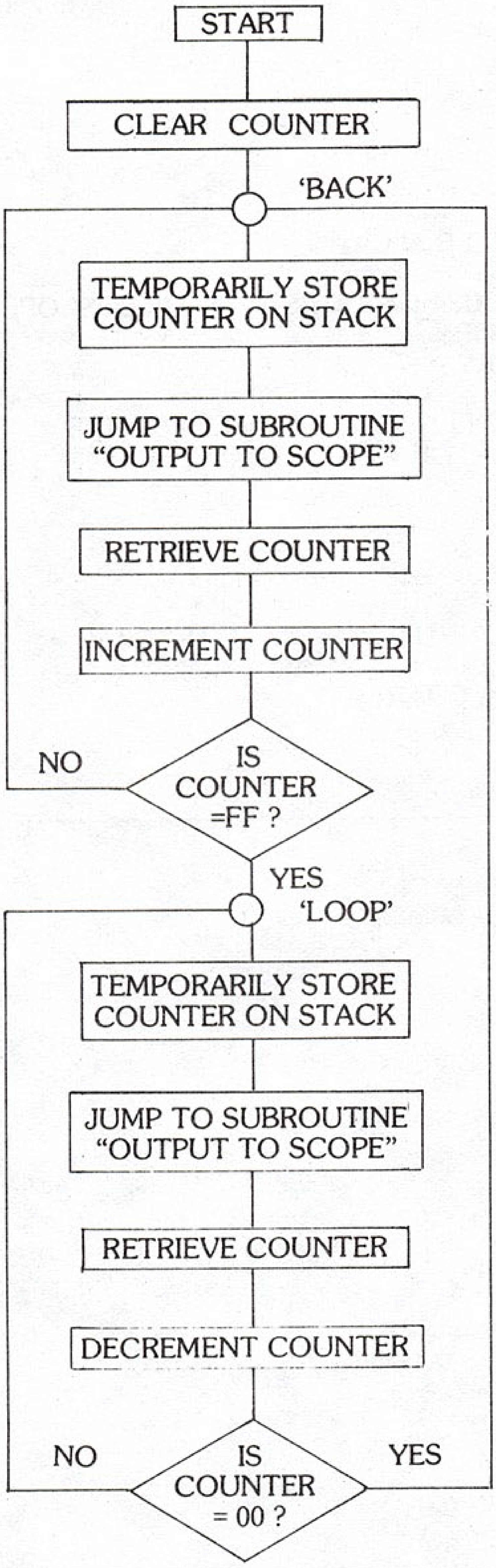
TABLE 7

Note that all of the registers except the Condition Code Register have their contents displayed as a decimal value. However, each bit within the Condition Code Register represents a flag which is either set to '1' or cleared, depending upon the arithmetic result of the previous instruction executed by the CPU. Therefore, it is most useful to display the contents of this CCR as eight binary digits on the Logger.

The only way to escape from this continuous looping display of the CPU registers is to press RESET. The user program may then be re-entered by defining program number '12' and by stepping FWD through the program codes, the SWI code can be replaced by the next executable opcode.

Section 4.4

USER PROGRAM PROJECT: TRIANGULAR WAVE OUTPUT TO OSCILLOSCOPE


START
CLEAR COUNTER
'BACK'
TEMPORARILY STORE COUNTER ON STACK
JUMP TO SUBROUTINE "OUTPUT TO SCOPE"
RETRIEVE COUNTER
INCREMENT COUNTER
NO
IS COUNTER =FF ?
YES
'LOOP'
TEMPORARILY STORE COUNTER ON STACK
JUMP TO SUBROUTINE "OUTPUT TO SCOPE"
RETRIEVE COUNTER
DECREMENT COUNTER
NO
IS COUNTER = 00 ?
YES

	Motorola Assembly Language	Memory Location	Decimal Codes Required
	CLRA	1	79
'BACK'	PSHA	2	54
	JSR $FE33	3,4,5	189,254,51
	PULA	6	50
	INCA	7	76
	CMPA$FF	8,9	129,255
	BNE'BACK'	10,11	38,246
'LOOP'	PSHA	12	54
	JSR $FE33	13,14,15	189,254,51
	PULA	16	50
	DECA	17	74
	BNE 'LOOP'	18,19	38,248
	BRA 'BACK'	20,21	32,236

FIGURE 10: FLOWCHART AND CODES FOR USER PROJECT

PROCEDURE

(a) RESET
(b) CALL PROGRAMME NUMBER 1 2
(c) ENTER
(d) REPLACE FIRST CODE BY PRESSING 7 9 AND ENTER
(e) PRESS > TO GAIN ACCESS TO NEXT MEMORY ADDRESS
(f) REPLACE NEXT CODE BY PRESSING 5 4 AND ENTER

and repeat for all 21 codes.

Section 4.4

SOME USEFUL SUBROUTINE MEMORY ADDRESSES

$F041	DISPLAY 'HELLO'
F176	OUTPUT 5 DIGIT VALUE TO DISPLAY
F21A	DELAY FOR 500 MILLISECONDS
F225	DISPLAY 'HI'
F27C	WAIT FOR 'START' PULSE
F287	FIND AVERAGE OF 256 SAMPLES
F2A0	MAKE AN ANALOGUE SAMPLE
F38E	DISPLAY 'LO'
F407	OUTPUT MEMORY AND CONTENTS TO DISPLAY
F538	OUTPUT 8 BITS OF ACCB ON DISPLAY
F62A	OUTPUT DAV PULSE FCA5 OUTPUT 1024dec BYTES TO OSCILLOSCOPE
F7D5	MOVE BACKWARDS THROUGH MEMORY
F7F5	MOVE FORWARDS THROUGH MEMORY
F609	SELECT INPUT ANALOGUE CHANNEL
F10B	OUTPUT POSITIVE SYNC STEP F9F9 DELAY FOR 50 MILLISECONDS
FD72	OUTPUT A SYNC PULSE FDEE CHECK FOR 1,2,3 OR 4 KEYPRESS
FE00	TEST FOR KEYPRESS
FE19	CLEAR ALL 4K RAM LOCATIONS
FE25	BRING ON 'VOLTS' LED
FE29	BRING ON 'SECS' LED
FE2D	BRING ON 'HERTZ' LED
FE33	OUTPUT TO OSCILLOSCOPE FE8E CONVERT BINARY TO DECIMAL
FED1	CONVERT DECIMAL TO BINARY
FF1C	BLANK THE DISPLAY FFB8 OUTPUT A CHARACTER
FF2A	INPUT '2 DIGIT PROG NUMBER'
FF52	INPUT '3 DIGITS AND ENTER'
FFC5	INPUT A CHARACTER

Section 4.5

TECHNICAL SPECIFICATION

Microprocessor	6802.
System RAM	4k byte of battery proteced CMOS RAM, allowing data retention own power down
Applications Software	Eprom based 4k byte
Expansion Software	12k byte, ie space for 3 additional application EPROMs
Analogue Inputs	4 analogue channels each with a switch selected range of +/–25 V, +/–2.5 V, +/–250 mV.
	8 bit resolution, +/– 1/2 LSB linearity A/D converter. Readings also subject to 1 LSB quantisation error. Input impedance 1 M ohm.
Pulse input	External: Trigger level: 1 volt Maximum input: 25 volts Input impedance: 1 M ohm
	Internal: Via channel 1 amplifier
Digital Input/Output Port	26 way IDC connector
	16 data lines, 4 control lines (TTL compatible), +5 V, and ground
	When configured as outputs, each output will drive
	1 TTL load
Sync Output	Output voltage swing: 0 to 4 volts
	Output impedance: 600 ohms
Analogue Output	Output voltage swing: –2.5 volt to +2.5 volt: 8 bit resolution +/–1/2 LSB linearity D/A converter.
	Output impedance 600 ohms.
Power	240 v mains adaptor supplied.
For remote operation	The logger requires 470 mA at 8 to 13 volt AC or DC.
Connections	All analogue I/0's via 4 mm sockets on a 0.75in spacing to take BNC adaptors Stock No. 456-009
Physical	Steel case 300 mm x 230 mm x 75 mm. Weight 2.2 kg.