FILE

INTERNAL DOCUMENT 247

I.O.S.

THE DATARING INTELLIGENT TIDE GAUGE SYSTEM REMOTE SITE

by

R I R PALIN

NATURAL ENVIRONMENT RESEARCH COUNCIL

INSTITUTE OF OCEANOGRAPHIC SCIENCES

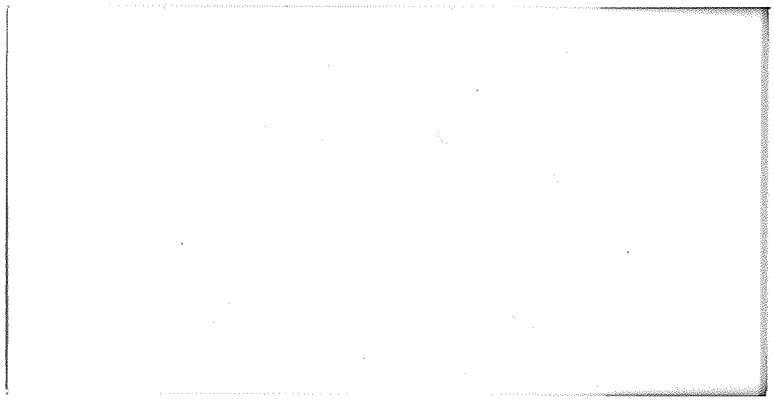
INSTITUTE OF OCEANOGRAPHIC SCIENCES

Wormley, Godalming,
Surrey GU8 5UB
(042-879-4141)

(Director: Dr. A. S. Laughton)

Bidston Observatory,
Birkenhead,
Merseyside L43 7RA
(051-653-8633)
(Assistant Director: Dr. D. E. Cartwright)

Crossway,
Taunton,
Somerset TA1 2DW
(0823-86211)
(Assistant Director: M. J. Tucker)



THE DATARING INTELLIGENT TIDE GAUGE SYSTEM REMOTE SITE

by

R I R PALIN

Internal Document No 247

Institute of Oceanographic Sciences
Bidston Observatory
Birkenhead, Merseyside

March 1986

CONTENTS

1. The Dataring system.

Dataring is a computer based logging and processing system which can be interrogated remotely by any 300 baud full duplex modem and terminal to transmit back data from up to seven sensors.

The design aims of the system are to provide a versatile logger, initially for use at remote tide gauge sites for logging tidal data, which consumes a small enough amount of power to be run for up to a fortnight on built in batteries.

The controlling element of the system is an Intel 8085 microprocessor with conventional Hmos ancillary support chips, and to obtain the low power, this is only switched on for a few milliseconds every time it is required to log the data from the sensors (the only exception to this being when a remote device is communicating with it when it is switched on for the duration of the call).

The 8085 based microcomputer, stores data from the sensors in 16 Kbytes of Random Access Memory (RAM), which is permanently powered up to retain the data, but consumes very little power when it is not in use. Communication, either with a remote device via the built in modem, or with a terminal plugged into the front panel, is handled by the

RS232 interface card.With the exception of this card and the mains power unit,all the system functions are implemented using COSMOS devices,and so can be powered up permanently to perform the routine functions of the logger.To prevent the cosmos circuits presenting potentially damaging voltages to the microcomputer whilst it is powered down,and also to adjust the voltages between the two different technologies,a buffer board is interposed between the processor and the cosmos circuits,which are all mounted on a parallel bus,referred to as the cosmos data bus,along which the signals to and from the cosmos cards travel.

The system comprises the following cards on the cosmos bus-

1) A real time clock card,for putting the time of a scan onto the logged data,as well as producing pulses at the required scan interval.

2) A start up card,which stores the source of any request to start up the sytem,and sends a start pulse to the switched power supply.

3) A status/DVM card,which enables the processor to read the state of up to seven digital single bit signals, And also reads the voltage of the stand-by battery.

4) Provision for up to seven sensor cards,which can be

configured for any digital or analogue sensor and convert its output to a signal in the required units using calibration equations stored on the boards.

Apart from these cosmos cards, and those associated with the processor as already mentioned, there is a power supply card for powering the system from the mains and in the event of a mains failure a relay on the board will switch in a set of stand-by batteries to power the system for up to a fortnight. The whole dataring system, together with a modem which will automatically answer incoming calls on the phone line, is housed in an environmental cabinet with a capacity for two 19" racks. All the dataring cards, apart from the PSU are housed on imperial standard, 4.5" x 8" cards with single or double sided 43 way edge connectors.

1.1 Switch on sequence.

The dataring is able to consume so little power because of the fact that the processor is on for only a very short time whilst it is logging a scan, usually only once every 15 minutes, and so considerable care is taken to ensure that the switch on is as fast and reliable as possible. The exact electrical performance of the system is detailed in the section giving the specification of the cards(q.v.), and so only an overview will be given here.

When a processor is switched on, it is customary to hold it in a reset condition for up to a second, to ensure that the power rails have fully stabilised, whereas it should be ready to run about 500 micro-seconds after the power has settled. The first system tried for powering up, detected when the power rails had stabilised, and then held the system reset for one milli- second. It was found, however, that this did not work reliably, and that the processor would start with its programme counter not at zero, meaning that programme execution would commence at some random address, and that the system would not work as required.

In the final design, the cosmos bus and the data stored in the cmos ram are not enabled until the processor has started correctly.

When the power supply to the processor board is off, the RAM and the cmos buffer are both electrically disabled. When

the pulse to turn on the power supply is applied, the reset rail to the processor is set high, and remains high, until the power rail reaches 97% of its final value. At this time the reset rail is dropped low, with the ram and cmos bus still disabled, for a period of 1 mS. If, during this time, the processor starts programme execution correctly at address zero, it will send out pulses to enable the cmos ram and databus, and also prevent further reset pulses being generated by the reset circuitry. If, however, the processor starts at some other address, then its behaviour will be random, but the chances of it enabling the cmos ram and thus corrupting the stored data, are less than one in ten to the thirty five, i.e. negligible. At the end of the one millisecond period, another reset pulse will be applied, and the process will repeat.

1.2 Interface to the Cmos cards.

As already stated, the cmos cards are buffered from the processor by a buffer board which performs three functions, viz-

1) Read data along an eight bit bus from the sensor cards.

2) Write data down an eight bit bus to the sensor cards.

3) Send commands to the cards along four sub-address lines to instruct them to read or write data.

Individual cards on the cmos bus, are selected by the buffer card by fifteen active low lines, ten of which go individually to the ten card slots in the card rack of the system. Additionally, the cmos bus contains extra lines carrying power rails and timing pulses, as detailed in the 'CMOS data bus' section.

1.3 Data modem.

The dataring is connected to a standard PSTN link via an auto-answer 300 baud, full duplex modem, type M321, made by Modular Technology. This is connected to the dataring RS232 board via a four wire link carrying the signal ground, TX data, RX data, and DCD (data carrier detect) lines. The DCD line is used to inform the system that a remote device is trying to establish contact with it. A terminal with the same four wires can be plugged in to perform the same functions as the modem.

2. Datarino power supply board

The power supply pcb provides all the power used by the system, and also houses various other small circuits. There are three different voltages available, +5 and -5 volts, both continuously present, and a switched +5 power rail which is used only whilst the processor is switched on. The miscellaneous functions are as follows-

1) Detect failure of the negative rail stand-by battery pack.

2) Detect correct function of the switched 5 volt supply, and when it occurs, enable the memory.

3) Provide the reset pulses to the microprocessor when the power is first applied, and disable them on command from the computer.

4) Provide a manual start pulse from a micro-switch, to turn the processor on via the start up board.

2.1 Main power supply.

The main power source is the toroidal transformer T1, which with capacitors C1 and C2 provides about +/- 12 volts. The relay RL1 with D7/C3 is closed as long as the transformer provides power, but in the event of a mains or transformer failure, the relay opens, and the stand-by battery pack supplies power to the two capacitors directly. The relay will switch in the batteries well before the voltage falls below acceptable limits, and so there is no interruption in the power supply from the stabilisers.

The primary voltage reference is provided by the band gap diode ZD1 and amplified up to 5 volts by IC2/TR2 to provide the continuous 5 volt supply. The exact voltage may be set using VR1. IC1/TR1 provides a gain of -1 from the stabilised 5 volt rail, and thus provides the continuous -5 volt rail.

2.2 Switched power supply.

On receipt of a pulse from the start up board(q.v.) at pin 24 the flip flop comprising D4-D6,R17/18 and IC8 goes into the 'set' condition,thus applying 5 volts to the input of the unity gain amplifier IC3/TR3.The 5 volt output provides the switched power supply to the processor.The power is switched off again by a pulse from the processor applied to pin 21(or from the manual stop switch on the board,uS2).

2.3 Negative supply fail detect.

IC4 and ZD2 together with the associated resistors,form a negative level detector,such that if the negative battery volts falls below about 7 volts the output of the op-amp goes high,and sets one of the status lines of the status card,which can be detected when the unit is interrogated.The exact voltage at which the unit operates is set by VR2.

2.4 Pulse power detect circuit.

IC5 is used as another level detector to indicate when the switched 5 volt rail has reached 97% of its final value.It is fed from a dedicated line from the back plane,originating at the point where the 5 volt rail is applied to the processor board,and in this way any delay along the power supply rails is allowed for

2.5 Reset halt/memory enable circuit.

When the power to the processor is first applied,the output rail from the invertor on the power detect circuit will be low,thus ensuring that the flip-flop comprising D1/2,R15/16 and IC6 is off.In this state the memory enable line will also be low,meaning that the Cmos store of the system cannot be accessed and possibly corrupted.At the same time the RS line of the 4040 IC7 will be low,and thus the 64 kHz pulses at pin 20 will produce a pulse at 1 mS intervals at the input of the and gate connected to D3 and pin 19.As soon as the voltage to the processor is sufficient to trigger the pulse power detect circuit,the power detect line will go high, and the 1 mS pulses will be allowed through the gate and apolied to the reset input of the microprocessor.On

receipt of a pulse at pin 17(reset halt in),or from the switch uS3,the flip flop will change state,thus halting the pulse train from IC7,and at the same time,the memory enable line will go high.Since the reset halt pulse is only generated by the processor when it has started up correctly,this means that the processor will be reset at 1 mS intervals until such time as it starts correctly,and until this happens, the memory is protected. The switch uS3 is provided so that the processor may be started manually for test purposes.

3. Processor board

Refer to the 8085 CPU drawing, and also manufacturers' data sheets.

The processor board is based on the Intel 8085 microprocessor, and contains four main modules, viz

1) The microprocessor.

2) The memory, consisting of 4 Kbytes of eprom, and 1 kbyte of RAM.

3) The USART, an 8251 for communication with RS232 devices.

4) The 24 bit I/O chip, type 8255 to control the Cosmos bus.

All the decoded address and the data lines, together with read, write, reset in and out, and two 8Kbyte page select lines, are brought out to the 43 way double sided edge connector on the component, or 'b' side of the board, whilst the 24 lines from the I/O chip are taken to the 'a' side.

3.1 The Microprocessor

The 8085 is fed from a 4.9152 Mhz. crystal as the clock, which is lower than the maximum speed at which the system will run, but this frequency has three benefits, viz-

1) The frequency can be reduced with the straight binary divider IC1 to give the correct frequencies for the standard baud rates to the USART IC4.

2) The reduced speed means that memories(paticularly the Cmos RAM) can be of the lower speed specification.

3) The lower clock speed leds to a slight reduction in the system power consumption.

The reset in to the chip is fed from a two input NOR gate IC6, which also changes the polarity of the reset input to positive, the reset out from the processor being fed to pin B27, for use by external circuits, and also to the 8251 IC4, and the 8255, IC12. The READY, TRAP, RESTART, and INTERRUPT lines are all tied to the relevant power rails through pull-up/down resistors, and may be wired to the spare pins on the edge connector if required in the future.

The multiplexed address/data lines are fed to the eight bit latch IC8(8282) and the address information is latched in by the ADDRESS-LATCH-ENABLE line on the processor(ALE pin 30). The output of the latch feeds the output from the board and also the address bus for use by the components on the

board.

3.2 Address decoding.

Address decoding is perforemd by IC3,IC2,and IC5,as follows-

IC3 does the decoding for the two IO chips,such that they are only selected during IO operations.The usart control and data addresses are EDh and ECh respectively.The 8255 has the control at 13h,with ports 'a' to 'c' at 10h-13h.

It should be noted that full address decoding is not used,either for the IO or the memory,and hence the various devices are also available at other locations within the memory ma.

IC2 decodes the top three address lines of the address bus,to give 8 8k 'pages' of memory.Page '0',from 0000h to 1fffh is used by the EPROMs IC11 and IC9,which are addressed only in the lower 4k of this page from 0000h to 0FFFh.The RAM IC7,which is a 1kx8bit device is 'echoed' 8 times in the second page of memory,and is addressed in the segment 3C00h to 3FFFh to maintain compatibility with the intel computer board,and any software written for it.

Pages 3 and 4 of the memory are fed to the edge connector for use by the CMOS rams in the data store,and give access to addresses 4000h to 7FFFh.

IC5 decodes address lines A11 and A12 to select the relevant 1k blocks within the first 8k page to be used by tthe EPROMS.

3.3 The USARI.

The 8251 usart IC4 is fed directly from the processor bus,and will operate at baud rates from 300 to 19.2kbaud by suitable choice of the link from the baud rate generator IC1.The unipolar output and input for transmit and Receive are fed off the board to be converted to bi-polar voltages for use by RS232 devices.The RX READY line is also brought out for use by the auto time-out circuit on another board.

4. Buffer board.

The dataring buffer board performs two main functions;firstly,it provides an interface between the 8255 IO chip on the processor board and the CMOS bus for the sensor cards,and secondly it decodes 4 of the IO rails to select one of ten cards in the card frame.

4.1 Read line buffer.

Data to be input to the processor is presented by the sensors etc. on lines 2 to 9 of the Cmos bus.To allow for low output levels,these signals are buffered using CD40109 level convertors(IC5 & IC8),whose Vcc inputs are fed from a 3 volt rail from IC2.These buffers are tri-state devices which are disabled when the processor power is turned off,by connection to the memory enable line.This ensures that no voltage is applied to the IO chips on the processor board when they are not powered up,since the presence of an input can destroy the device.

4.2 Write line buffer.

The write lines are buffered to Cmos levels with HEF4050 buffers whose inputs are connected to pull up resistors to ensure that the output of the 8255 IO chip is sufficient to switch them.Since no card will ever be selected when the processor is powered off,there is no need to disable the buffers at any time.

4.3 Card select decoder.

The cards are selected by a four-to-sixteen line decoder with active low outputs which is enabled,via an inverter,from the memory enable line(IC6). The low three inputs to the decoder are fed via pull-up resistors,but the high line on pin B25 does not have a pull up,since this line is used during the power up sequence to provide a positive pulse once the IO chip is set up ,and if pull up resistors are used on the line,the pulse will appear immediately the device receives power.

4.4 Time out reset circuit.

IC3 and the two gates of the IC1 connected to it, form a timer which puts a reset pulse into the processor board if no input is received by the processor from the USART for 128 seconds.

If the memory enable line is low, i.e. the processor is powered off, then the RS input to the 4040 IC3 is unconditionally high and no pulses will appear on the output of the chip at pin 13. If, however, the memory enable is high, then the one second pulses at the clock input at pin A26 will be counted up to 128 when a pulse will appear at the output pin 13. Pin B36 is connected to the RX ready line of the USART on the processor board, which goes high whenever a character is received from the RS232 interface. Thus, the processor is only ever reset from this device if the system receives no input from the modem (or keyboard) for 128 seconds, which it will take to mean that contact has been lost. The resistor, capacitor and diode serve to differentiate the change in level to give a short pulse.

5. Start up board.

Refer to 'Dataring start up board' drawing.

The dataring start up board is responsible for providing a pulse to the switched power supply(q.v.) whenever the processor needs to be powered up.At the moment,this can be from four different sources,viz-

1) A pulse from the clock card to indicate that the sensors have just been updated,this initiates a scan.

2) The auto-answer modem has detected a call,or a terminal has been plugged into the front panel connector.

3) One of the two datum probes has switched.

4) The visit button has been depressed.

In addition to starting the processor,the card also contains the circuitry to provide the stop pulse,and a timer which can be reset and read by the processor,to provide a 64 second interval to be used for timing out when a line is being input to the system from the terminal/modem.

5.1 Start-up circuits.

The start up circuits for the four sources of interrupt operate in the same way, with the exception that whereas the clock and terminal/modem will start the system with a short duration pulse, the datum probes and visit switch must provide a voltage for about one second before the system will operate, due to the presence of the delay network on the input schmitt trigger.

The inputs from the modem and terminal DCD lines, which appear on pins A41 and A39 are presented to the diode 'OR' gate D3, D4 and R10, since they are treated identically by the processor, and this line, together with the other four, is differentiated by the RC networks RP1 etc., and presented to the set input of the tri state quad RS latch IC5/7 so that the source of the interrupt can be detected by the processor. The seven interrupt lines are also input to the 8 input NOR gate IC4, the output of which is inverted to present a positive pulse to the start-up line on the power supply board.

The four to sixteen line decoder IC6 enables the two latches on receipt of the SA0 command, so that the tri state latches, which are in effect the interrupt register, can be read by the processor. The SA1 to SA7 commands will reset the interrupt register lines independantly to 'clear' the interrupts.

5.2 Time out timer.

The binary counter IC2 is fed from the cmos bus one second pulse via pin A26, and when a count of 64 seconds is reached, it sets the most significant bit of the interrupt register. The counter can be reset at any time by the SA8 command which puts a pulse onto the counter's MR line.

5.3 Power down circuit.

The SA15 command will put a pulse onto the monostable comprised of D1/2, R1/2/3, and C1 with the three inverting buffers of IC1, which will put a 0.1 second pulse out at pin A42. This pulse is fed to the 'STOP' line on the psu card, and turns off the power to the processor.

6. Clock card.

Refer to 'Dataring clock/sample-pulse board' drawing.

The clock card performs three distinct functions, viz-

1) Generation of xtal controlled frequencies for use by various other cards.

2) Provision of a real time clock which can be read by the micro-processor.

3) Control of data logging at intervals set by the micro.

All the frequecies used by the card are generated by a xtal oscillator based around a 2.097152 Mhz. xtal and an inverting buffer IC6. The frequency of the oscillator is set by a twisted wire pair C9, whose exact value is set by cutting the wire to obtain the correct frequency. The buffer on the output of the oscillator is neccessary to ensure that connection of the counter probe does not change the frequency of the circuit under test. This signal is divided down by the counter IC3 to provide three frequencies and an outout to the next divider stage.

The 32.768 Khz. signal is fed to the input of the real time clock chip to provide an accurate substitute for the 32

Khz. xtal that it is designed to use.

The 16.384 Khz. signal is fed to the power supply board and is used by it to provide the reset pulses at power-up(for more details see the relevant section).

The 4.096 Khz. signal is fed to the status/dvm board,and provides the clock for the dvm..

The second divider stage,IC2,provides a one second pulse to the Cosmos bus for general purpose use by the sensor cards,and also a half second pulse for the same bus and the divider IC1 which generates pulses at intervals of 1.875 minutes for sampling control,as follows.

When a count of 225 is reached on the counter IC1,which will happen 1.875 minutes after reset,the outputs 00,05,6,7, of the counter all go high together, which allows the voltage at the junction of R1 and C4 to rise from about 0.6 volts to 5 volts.This provides a clock pulse to IC4,and also forces the output of the 4072 IC7 high.After about 5 micro-seconds the voltage at the junction of R2 and C5 is sufficient to reset the counter to zero,while the RC network R1,C4 ensures that the reset pulse is long enough to correctly reset the counter.The second input to the OR gate is used to reset the counter when the sampling interval is changed,so that the samples are synchronised to the time at which the sampling interval is set.

Counter IC4 generates pulses at the four sample intervals used by the system,the interval needed being controlled by

an output of the 4508 IC11 and the four and gates IC10 together with the or gate IC7.A load pulse for the sensor cards is generated by C16/R6/IC5 and has a length of about 5 microseconds. the reset pulse occurs about 30 microseconds later and has a similar length, being generated by R8/C12/IC5.

6.1 Sub-address decoder IC8.

All four sub address lines are used by the clock card, the functions being as follows

SA 0	Chip select and read.
SA 3	Chip select only.
SA 7	Write and chip select.
SA 9	Clock data strobe.
SA 12	Sample interval strobe.

6.2 Real time clock chip.

Refer to the manufacturers data sheet.

As stated earlier, the 32768 Hz. clock for the chip is provided by the main crystal oscillator, so the other functions of the board are to read data from the clock and to write data to it.

6.3 Reading the clock.

This is the more complex of the two functions, in view of the stringent timing requirements for correct reading The address data is presented direct to the clock chip via the cosmos write bus, after this the read is executed by putting out sub address 0 to the board. The chip select line goes low immediately the sub address is written, and stays low until it is changed. The read line to the chip is supplied with a pulse about 8 micro-seconds delayed from the chip select pulse by about 100 nano-seconds under the control of the two RC networks R9/C18 and R10/C17. When the read line goes low, the data at the address presented from lines 10-13 appears on the low four lines of the cosmos read bus, and the processor has about 8 micro-seconds in which to read the data.

6.4 Writing to the clock.

Since there are no limitations on how long may be spent writing data to the clock, this may be done simply.

Data to be written is presented to the latch IC11 and latched in using sub-address 9. After the data is latched in, the address is presented on the write lines, and the data is written in by executing SA3, SA7, and finally SA3 which transfers the data into the clock.

6.5 Setting the sample interval.

This is done in exactly the same way as data is stored for writing to the clock, but with the execution of SA12 to select the other half of the latch.

7. DVM and status board.

Refer to 'Dataring DVM/status board' drawing.

The dvm/status board provides two functions. On receipt of the SA0 command it will give the voltage of the positive battery supply, and the SA1 command will give the state of the eight binary input lines, which are used to indicate the state of such things as the gas supply pressure on a GO/NO-GO basis. Because of limitations on board pins only lines 0 to 6 are available for use off the board, and of these line 6 is used to indicate negative supply fail, whilst line 5 indicates the presence of the DCD signal on the RS232 board.

7.1 DVM circuit.

The output of the DVM is an eight bit byte, with a resolution of 0.1 volts, so that the readings are in the range 0 to 25.5 volts.

The voltage to be measured is applied at pin A40 and thence through the voltage divider to the negative input of op-amp IC1. A clock signal of 64 kHz is applied to the CL input of the counter IC4 whose outputs are fed to the R/2R ladder network comprising R1 to R16. The effect of this

network is to feed a ramp voltage to the non-inverting input of the op-amp IC1,whose value is directly proportional to the binary number appearing on the output of the counter IC4.

When the voltage from the ramp generator just exceeds the input voltage from the divider,the output from the op-amp goes high,and provided the line on the and gate gate at the output is also high,a pulse will load the binary value from the counter IC4 into the latch IC6,where it may be read by the SA0 command.

The gate on the output of the op-amp is connected to the output of the decoder IC3,and is necessary to ensure that the latch cannot be updated whilst the voltage is being read by the processor.

7.2 Status register.

The eight status lines are fed to the resistor pad RP1,and the tri-state latch IC5,which is wired to be transparent to the input signals,so that when decoder IC3 receives the SA1 command,the state of the input of the latch can be read by the processor.

8. RS232 interface board.

The RS232 interface board converts the signals on the transmit, receive and DCD(data carrier detect) lines from the bi-polar voltages of the RS232 standard, to the unipolar voltages for the 8251 USART on the processor board.

In addition, the switch SW1, mounted at the front of the card, feeds the schmitt trigger through the RC network to provide a reset pulse to the microprocessor via pin A26 on the edge connector.

8.1 Transmit buffer.

The /TX line from the USART on the processor card is applied via pin A35 to an inverting amplifier, IC3. The output of this amplifier, which is not less than +- 3 volts, is applied to the front panel cannon connector pin 3, and also to the modem wires via pin B38.

8.2 Receive buffer.

Bi-polar signals with an amplitude of not less than +- 3.5 volts, are applied to resistors R1 and R10 from the cannon connector pin 2, and the modem via pin B39, and thence through resistors R2 and R11 respectively to two inputs of the NOR gate IC2. The gate protection diodes within the HEF4002 in conjunction with these series resistors serve to protect the gates from negative voltages, or voltages exceeding 5 volts, on the inputs. Thus signals on the receive lines are inverted and converted to uni-polar 5 volt levels to be applied to the /RX input of the USART.

8.3 DCD buffer.

The circuitry between pins D20 on the cannon connector/B40 from the modem, functions identically to the RX buffer except that the schmitt trigger is used to re invert the polarity of the signal which is applied to pin A41 and thence to the start up board where it starts up the processor when the modem receives an incoming call or a terminal device is plugged into the front panel connector.

9. Sensor boards.

Refer to "Dataring sensor board" drawing.

The sensor boards for the system are in two sections on one board. The first section is specific to the paticular sensor, and is responsible for converting the signal from the sensor to a frequency roughly in the range 0-4 Khz. which is then integrated and stored by the second half of the circuit, which is common to all boards.

A signal in the range 0-4Khz. is presented at the clock input of the 4040 IC3 and the signal divided down by 8, 16, 32 or 64 is presented to 4 'and' gates 4081. The other inputs to the and gates appear at the board edge conn- ector at pins A31 to A34, only one of which will be at logic '1' dependant on the integration period selected, thus the number of output pulses appearing at the output of 'or' gate IC6 4072, is independent of the integration period selected.

The signal from IC6 feeds two counters in series, IC2 & IC1. This gives a sixteen bit number which is presented to two 8-bit Tri-state latches IC5 and IC4. At the end of the integration period selected a load pulse appears at pin A27 and the data presented to the latches is stored. Shortly after the load pulse (about 10 microseconds), the reset pulse appears at pin A28, and the counters IC2 and IC1 are reset to zero ready for the next integration.

The stored 16 bit number can now be read by the microprocessor,8 bits at a time via the COSMOS bus read lines(A2 to A9).

The latches put their data onto the read lines when the /E input(pins 3&15) is forced low.The processor does this by first selecting the board with a low level on pin A38 presented to the /OE input of the 2-4 line decoder IC8. The sub address lines SA0 and SA1 at pins A18 and A19 will now read out the latches IC5 and IC6 at sub-address '0' and '1' respectively,these being the least and most significant data bytes.

The 2716 EPROM,IC7,stores the calibration coefficients for the individual sensor.Only address lines A0 to A7 are used,these being connected to the cosmos bus 'write' lines on pins A10 to A17.Addresses A8 to A10 are tied to ground,so that a total of 256 bytes is readable on the eprom.

It is selected by sub-address '3' on IC8,and when the switched 5 volt power supply is applied via pin A36,the data can be read on the cosmos read bus.

9.1 The Digiquartz interface board.

Refer to "Digiquatz Interface" drawing.

The digiquartz pressure sensor gives a frequency output in the range about 36-40 Khz, and since this is too high to be read by the sensor circuit, it must first be reduced. One solution would be to divide the input frequency by ten, but this would also reduce the resolution available. The solution adopted in the final design is to subtract a constant frequency from that input.

The sensor is powered from the +12 volt rail via a 50mA fuse and pin B42. The input frequency appears at pin A41, and passes through R2 to be presented to the clock input of the 4040 IC10, and at the input to the gate comprising IC12C and IC12A. A low level at the anode of diode D6 prevents the signal passing through the gate and appearing at the output of IC12A, whilst a high level allows the signal to pass.

A 5 micro-second pulse appears on pin A29 at half second intervals and this is used to reset the two 4040 counters IC10 and IC11, as well as to force a low level on the output of the flip-flop comprising IC12F and IC12B. Thus when the pulse appears the signal from the sensor cannot pass through the gate. When the outputs from 01, 02 & 06 of the 4040 IC11

all go high,a pulse is applied,via IC12D and IC12E,to the flip-flop 'set' input,and the sensor signal can pass through to the counters in the rest of the board until the next half-second pulse appears.

The effect of all this is to reduce the input frequency of the Digiquartz by 35840 Hz,whilst leaving the resolution unchanged.

9.2 The Ott sensor interface.

Refer to the "Ott Interface" drawing.

The Ott float gauge is attatched to a precision multi-turn potentiometer with a resistance of 5 K-ohms,and it is neccessary to convert this to a frequency.On the prsent sensor board this is done by a 9400CT voltage-to- frequency convertor and a precision voltage reference.

Band gap diode D1 is fed via resistor R10 from the +5 volt rail of the Cosmos data bus,and provides a voltage reference of about 1.2 volts,this is amplified by op-amp IC12 and feedback network R8/R9 to provide about 2.5 volts nominal supply to the potentiometer high end via pin B42.The output from the pot,which is proportional to the tidal elevation,appears at pin A42 and is presented to the amplifier comprising IC11,R6 R7 and VR2.The amplifier is set to have a

gain of less than two,and the trimmer VR2 should be set to give an output voltage at pin 6 of IC11 of 4.8 volts (+-.05) when the pot is at its highest elevation.This check can be carried out without the pot in place by joining pins A42 and B42 together.

The voltage resulting from the amplifier and pot. is presented to the input of the V-F convertor IC10.The output frequency of this chip is determined by several of the passive components connected to it,the only none-critical component being R2,which is just a load for the final frequency divider stage inside the chip.The trimmer VR1 sets the output frequency of the oscillator and should be set to give a frequency of about 10 Hz. at pin 10 of IC10 with the pot at the lowest elevation.Again this test can be carried out without the pot by shorting pin A42 to the 0 volt rail on the board.

APPENDIX 1

Microprocessor Bus.

1		+5 volts
2		D7
3		D6
4		D5
5		D4
6		D3
7		D2
8		D1
9		D0
10		A0
11		A1
12		A2
13		A3
14		A4
15		A5
16		A6
17		A7
18		A8
19		A9
20		A10
21		A11
22		A12
23	nc	
24		/WT
25		/RD
26		reset in
27		/RS out
28		
29		
30		
31		
32		
33		
34		
35		
36		
37	pol. key	
38		Page 1
39		Page 2
40		
41		
42		
43		ground 0 volts.

APPENDIX 2

Cosmos Data bus.

1	+5 volts continuous
2-9	Read lines (into 8085)
10-17	Write lines
18	Sub-Address 0
19	SA1
20	SA2
21	SA3
22	+12 volts unsmoothed
23	
24	
25	
26	1 second interval pulse
27	clock load pulse
28	clock reset pulse
29	0.5 second pulse
30	-12 volts
31	
31	Interval select 1.875 min.
32	IS 3.75 min.
33	IS 7.5 min.
34	IS 15.0 min.
35	Enable line
36	Switched +5 volts.

APPENDIX 3

Software_listing.

Dataring software assembler listing,produced using RIP/I85X 8085 cross assembler.

```
1                       ;  VERSION 1 OF DATARING SOTWARE
1                       TITLE   'Dataring v2.4'
1                       ;
1                       ;
1                       SIS:  EQU 4025H
1                       EXCF:  EQU 3EA0h
1                       CTRL:  EQU 13H
1                       PA:  EQU 10H
1                       PB:  EQU 11H
1                       PC:  EQU 12H
1                       UC:  EQU 0EDH
1                       UD:  EQU 0ECH
1                       STACKT: EQU  3CFAH
0  0000 3E90                PWUP:  MVI A,90H       ;SET PIO PORT
0  0002 D313                       OUT CTRL
0  0004 3ECF                       MVI A,0CFH     ;AND USART
0  0006 D3ED                       OUT UC
0  0008 3E25                       MVI A,25H       ;SET FOR ASYNCH
0  000A D3ED                       OUT UC
0  000C 3EFF                       MVI A,0ffh
0  000E D312                       OUT PC
0  0010 31FA3C                     LXI SP,STACKTOP
0  0013 C32700                     JMP RIS
0  0016 3E2F                PWDN: MVI A,2FH     ;TURN OFF THE PROCESSOR
0  0018 D312                       OUT PC
0  001A 76                         HLT
0  001B DBED                CI:    IN  UC ;    GET A CHAR FROM THE USART
0  001D E602                       ANI 02
0  001F CA1B00                     JZ  CI        ;RETURN IN THE ACCUMULATOR
0  0022 DBEC                       IN  UD
0  0024 E67F                       ANI 7FH       ;GET RID OF THE PARITY BIT
0  0026 C9                         RET
1                       ;
1                       ;
1                       ;  FIND OUT WHAT WOKE THE PROCESSOR UP
1                       ;
1                       ;
0  0027 CD3C01           RIS:  CALL DLY1     ;GIVE THINGS A CHANCE TO SETTLE
0  002A 3E20                   MVI  A,20H
0  002C D312                   OUT  PC
0  002E DB10                   IN   PA
0  0030 E61F                   ANI  1FH
0  0032 CA1600                 JZ   PWDN     ;SO GO BACK TO SLEEP
0  0035 322540                 STA  SIS      ;SAVE WAKE UP CODE
0  0038 E601                   ANI  01H      ;IS IT THE RS232 INTERFACE?
0  003A CA9200                 JZ   RIS2
0  003D 3E21                   MVI  A,21H    ;RESET THE FLAG
0  003F D312                   OUT PC
0  0041 CD1B00           RIS1: CALL CI       ;COMMS SEGMENT
0  0044 FE0D                   CPI  0DH      ;WAIT FOR TWO CARRIAGE RETURNS
0  0046 C24100                 JNZ  RIS1
0  0049 CD1B00                 CALL CI
0  004C FE0D                   CPI  0DH
0  004E C24100                 JNZ  RIS1
0  0051 3AA03E                 LDA  EXCF
0  0054 CD2F07                 CALL APRT
```

```
0  0057 0A0D                DW   0D0AH
0  0059 4249445354   DB   'BIDSTON DATARING  v 2.4'
0       4F4E204441
0       544152494E
0       4720207620
0       322E34
0  0070 0A0D                dw   0d0ah
0  0072 4269647374   db   'Bidston test set     '
0       6F6E207465
0       7374207365
0       7420202020
0  0073 04                  DB   04H
0  0087 CDD100              CALL PSWD
0  008A FEFF                CPI  0FFH
0  008C CA2700              JZ   RIS
0  008F CD0008       PMT:   CALL PMPT
0  0092 3A2540       RIS2:  LDA  SIS
0  0095 E602                ANI  02H
0  0097 CAA100              JZ   RIS3
0  009A 3E22                MVI  A,22H
0  009C D312                OUT  PC
0  009E CD000C              CALL LGSC
0  00A1 3A2540       RIS3:  LDA  SIS
0  00A4 E604                ANI  04H
0  00A6 CAB000              JZ   RIS4
0  00A9 3E23                MVI  A,23H
0  00AB D312                OUT  PC
0  00AD CDE60D              CALL DPIR
0  00B0 3A2540       RIS4:  LDA  SIS
0  00B3 E608                ANI  08H
0  00B5 CABF00              JZ   RIS5
0  00B8 3E24                MVI  A,24H
0  00BA D312                OUT  PC
0  00BC CDCE0D              CALL DPR
0  00BF 3A2540       RIS5:  LDA  SIS
0  00C2 E610                ANI  10H
0  00C4 CACE00              JZ   RIS6
0  00C7 3E25                MVI  A,25h
0  00C9 D312                OUT  PC
0  00CB CD4E0E              CALL LGVS
0  00CE C32700       RIS6:  JMP  RIS
0  00D1 CD9101       PSWD:  CALL CRLF
0  00D4 1603                MVI  D,03H
0  00D6 CD9101       PSW1:  CALL CRLF
0  00D9 CD2F07              CALL APRT
0  00DC 5041535357   DB   'PASSWORD?'
0       4F52443F
0  00E5 0A0D                DW   0D0AH
0  00E7 2A04                DW   042AH
0  00E9 CD4301       PSW2:  CALL BIL
0  00EC FE0D                CPI  0DH
0  00EE CAF700              JZ   psw3
0  00F1 CD9101              call  crlf
0  00F4 C3E900              jmp  psw2
0  00F7 21C03D       PSW3:  LXI  H,3DC0H
0  00FA D5                  PUSH D
```

```
0  00FB 113201          LXI  D,PSWM
0  00FE 1A        PSW4: LDAX D
0  00FF FE04            CPI  04H
0  0101 CA2F01          JZ   PWOK
0  0104 BE              CMP  M
0  0105 C20D01          JNZ  PWWR
0  0108 23              INX  H
0  0109 13              INX  D
0  010A C3FE00          JMP  PSW4
0  010D 21C03D    PWWR: LXI  H,3DC0H
0  0110 113601          LXI  D,PSXC
0  0113 1A        PSW5: LDAX D
0  0114 FE04            CPI  04H
0  0116 CA2A01          JZ   XCOK
0  0119 BE              CMP  M
0  011A C2A701          JNZ  hight
0  011D 23              INX  H
0  011E 13              INX  D
0  011F C31301          JMP  PSW5
0  0122 D1        PWR1: POP  D
0  0123 15              DCR  D
0  0124 C2D600          JNZ  PSW1
0  0127 3EFF            MVI  A,0FFH
0  0129 C9              RET
0  012A 3E01      XCOK: MVI  A,01
0  012C 32A03E          STA  EXCF
0  012F D1        PWOK: POP  D
0  0130 AF              XRA  A
0  0131 C9              RET
0  0132 0000      PSWM: DW   0H    ;   insert the right passwords here.
0  0134 0000            DW   0H
0  0136 0000      PSXC: DW   0H
0  0138 0000            DW   0H
0  013A 0000            DW   0H
0  013C 3E0F      DLY1: MVI  A,0FH
0  013E 3D        DL1:  DCR  A
0  013F C23E01          JNZ  DL1
0  0142 C9              RET
0  0143 3E28      BIL:  MVI  A,28H
0  0145 D312            OUT  PC
0  0147 3E20            MVI  A,20H
0  0149 D312            OUT  PC
0  014B 21C03D    BIL1: LXI  H,3DC0H
0  014E DB10      BIL2: IN   PA
0  0150 E680            ANI  80H
0  0152 FE80            CPI  80H
0  0154 CC0E0D          CZ   CKSC
0  0157 DBED            IN   UC
0  0159 E602            ANI  02H
0  015B CA4E01          JZ   BIL2
0  015E DBEC            IN   UD
0  0160 E67F            ANI  7FH
0  0162 FE7F            CPI  7FH
0  0164 C27601          JNZ  BIL3
0  0167 3EC0            MVI  A,0C0H
0  0169 BD              CMP  L
```

```
0  016A CA4E01              JZ   BIL2
0  016D 0E2A                MVI  C,2AH
0  016F CD9C01              CALL CO
0  0172 2D                  DCR  L
0  0173 C34E01              JMP  BIL2
0  0176 77           BIL3:  MOV  M,A
0  0177 2C                  INR  L
0  0178 CA8601              JZ   EROR
0  017B FE0D                CPI  0DH
0  017D C8                  RZ
0  017E FE1B                CPI  1BH
0  0180 CA8601              JZ   EROR
0  0183 C34E01              JMP  BIL2
0  0186 0E21         EROR:  MVI  C,21H
0  0188 CD9C01              CALL CO
0  018B CD9101              CALL CRLF
0  018E C34B01              JMP  BIL1
0  0191 0E0A         CRLF:  MVI  C,0AH
0  0193 CD9C01              CALL CO
0  0196 0E0D                MVI  C,0DH
0  0198 CD9C01              CALL CO
0  019B C9                  RET
0  019C DBED         CO:    IN   UC
0  019E E601                ANI  01H
0  01A0 CA9C01              JZ   CO
0  01A3 79                  MOV  A,C
0  01A4 D3EC                OUT  UD
0  01A6 C9                  RET
0  01A7 21C03D       hight:  lxi  h,3dc0h
0  01AA 110D02               LXI  D,psht
0  01AD 1A           h1:  ldax  d
0  01AE FE04                cpi 04
0  01B0 CABC01              jz h4
0  01B3 BE                  cmp  m
0  01B4 C22201              jnz  pwr1
0  01B7 23                  inx  h
0  01B8 13                  inx  d
0  01B9 C3AD01              jmp  h1
0  01BC CD2F07       h4:  call aprt
0  01BF 4C61737420   db  'Last recorded height-  '
0       7265636F72
0       6465642068
0       6569676874
0       2D2020
0  01C0 04                  db  04h
0  01D7 3A1140              lda ptr1
0  01DA 67                  mov  h,a
0  01DB 3A1240              lda  ptr2
0  01DE 6F                  mov  l,a
0  01DF CDE801              call bakup
0  01E2 CDFC05              call tasc
0  01E5 C31600              jmp pwdn
0  01E8 D5           bakup: push  d
0  01E9 E5                  push h
0  01EA 16FF                mvi d,0ffh
0  01EC 15           bak1:  dcr  d
```

```
0  01ED CA0902              jz  nobak
0  01F0 2B                  dcx  h
0  01F1 7E                  mov  a,m
0  01F2 FEA3                cpi 0a3h
0  01F4 C2EC01              jnz  bak1
0  01F7 23                  inx  h
0  01F8 23                  inx  h
0  01F9 23                  inx  h
0  01FA 23                  inx  h
0  01FB 7E                  mov  a,m
0  01FC FEF2                cpi  0f2h
0  01FE 2B                  dcx  h
0  01FF 2B                  dcx  h
0  0200 2B                  dcx  h
0  0201 2B                  dcx  h
0  0202 C2EC01              jnz  bak1
0  0205 D1                  pop  d
0  0206 D1                  pop  d
0  0207 B7                  ora  a
0  0208 C9                  ret
0  0209 E1           nobak:  pop  h
0  020A D1                   pop d
0  020B 37                   stc
0  020C C9                   ret
0  020D 4845         psht:   dw  4548h
0  020F 49474854      db  'IGHT'
0  0210 04                   db  04
1                ptr1:  equ   4011h
1                ptr2:  equ   4012h
1                memt:   equ  80h
1
;  TR segment of software.Sends back data in various formats
1                ;  originally assembled from 0400h.
1
;  when entered from command interpreter,it issues a '>' prompt
1
;  character and waits for a number to select an option.Any
1                ;  non-numeric input is illegal.
1                esc:  equ  1bh
1                sp:   equ  20h
1                cr:   equ  0dh
1                PSV1:  EQU 4013H
1                PSV2:  EQU 4014H
1                PSV3:  EQU 400EH
1                PSV4:  EQU 400FH
1                STRBOT: EQU 41H
1                SCARD:  EQU 03H
1                bnfl:  equ  4015h
1                strtop:  equ 60h
o1                ORG  1024
0  0400 0E3E         TR: MVI  C,3EH
0  0402 CD1709              CALL c0 ;  SEND A PROMPT TO USART
0  0405 CD0E0D       TR1: CALL CKSC  ;  LOOK FOR FLAG FROM SENSORS
0  0408 DBED                IN   UC
0  040A E602                ANI  02H
0  040C CA0504              JZ   TR1  ;   WAIT FOR CHAR FROM USART
```

```
0  040F DBEC              IN   UD
0  0411 E67F              ANI  7FH  ;   STRIP OFF PARITY
0  0413 FE31              CPI  31H  ;   CHECK INPUT AND GO TO LABEL
0  0415 CA3904            JZ   TRL1
0  0418 FE32              CPI  32H
0  041A CA4704            JZ   TRL2
0  041D FE33              CPI  33H
0  041F CA6304            JZ   TRL3
0  0422 FE34              CPI  34H
0  0424 CA7104            JZ   TRL4
0  0427 FE35              CPI  35H
0  0429 CA8004            JZ   TRL5
0  042C FE30              CPI  30H
0  042E CA0008            JZ   PMPT  ;  GO BACK TO COMMAND IF '0'
0  0431 0E21              MVI  C,21H
0  0433 CD1709            CALL c0 ;  ILLEGAL CHAR,SO OUTPUT '!'
0  0436 C30004            JMP  TR   ;  TRY AGAIN
0  0439 3A1340      TRL1: LDA  PSV1 ;  SEND BACK DATA IN ASCII
0  043C 67
MOV  H,A  ;  PUT ADDRESS TO START TRANSMISSION IN HL PAIR
0  043D 3A1440            LDA  PSV2
0  0440 6F                MOV  L,A
0  0441 CDFC05            CALL TASC ;  GOTO ASCII TRANSMIT SEGMENT
0  0444 C30004            JMP  TR
0  0447 3A1140      TRL2: LDA  PTR1 ;  SEND BACK DATA FROM  MIDNIGHT
0  044A 67                MOV  H,A  ;  PUT CURRENT DATA POINTER INTO HL
0  044B 3A1240            LDA  PTR2
0  044E 6F                MOV  L,A
0  044F CDC605            CALL FMDN ;  NOW GET MIDNIGHT ADDR. INTO HL
0  0452 C25D04            JNZ  TR21 ;  ACC. SET TO ZERO IF NOT SUCCESFUL
0  0455 0E21              MVI  C,21H
0  0457 CD1709            CALL c0
0  045A C30004            JMP  TR
0  045D CDFC05      TR21: CALL TASC
0  0460 C30004            JMP  TR
1                  ;
1                  ;
0  0463 3A1340      TRL3: LDA  PSV1  ;  SEND BACK DATA IN PSUEDO BINARY
0  0466 67                MOV  H,A
0  0467 3A1440            LDA  PSV2
0  046A 6F                MOV  L,A
0  046B CD4B05            CALL TBIN
0  046E C30004            JMP  TR
1                  ;L
1                  ;
0  0471 CD0E0D
TRL4: CALL CKSC  ;  SET TRANSMIT POINTER FROM CHAR SENT TO
0  0474 DBED              IN   UC    ;   USART
0  0476 E602              ANI  02H
0  0478 CA7104            JZ   TRL4
0  047B DBEC              IN   UD
0  047D C3BA07            jmp  ltr
1                  ;
1                  ;
0  0480 CD2F07      TRL5: CALL APRT
0  0483 464C414753   DB   'FLAGS'
```

```
0  0488 0A0D                DW   0D0AH
0  0489 04                  DB   04H
0  048B 211040              LXI  H,4010H
0  048E 1610                MVI  D,10H
0  0490 7E           TR51:  MOV  A,M
0  0491 CD4107              CALL PRAC
0  0494 CD2907              CALL SPCE  ;  SEND OUT FLAG DATA
0  0497 23                  INX  H
0  0498 15                  DCR  D
0  0499 C29004              JNZ  TR51
0  049C CD0C09              CALL crlf2  ; RET AT END OF LINE OF 16 FLAGS
0  049F 1610                MVI  D,10H  ; RESET D FOR NEXT LINE
0  04A1 7E           TR52:  MOV  A,M
0  04A2 CD4107              CALL PRAC
0  04A5 CD2907              CALL SPCE
0  04A8 23                  INX  H
0  04A9 15                  DCR  D
0  04AA C2A104              JNZ  TR52
0  04AD CD0C09              CALL crlf2
0  04B0 CD0C09              CALL crlf2
0  04B3 CD2F07              CALL APRT
0  04B6 5354415455    DB    'STATUS  '
0       532020
0  04B7 04                  DB   04H
0  04BF 3E31
MVI  A,31h  ;  PUT IN THE ADDR. OF THE STATUS CARD CH1
0  04C1 D312                OUT  PC
0  04C3 DB10                IN   PA
0  04C5 CD3205              CALL PBIN  ;  PRINT THE STATUS WORD IN BINARY.
0  04C8 CD0C09              CALL crlf2
0  04CB CD0C09              CALL crlf2
0  04CE CD2F07              CALL APRT
0  04D1 4241545445    DB    'BATTERY VOLTS= '
0       525920564F
0       4C54533D20
0  04D2 04                  DB   04H
0  04E1 3E30                MVI  A,30h  ;  READ THE DVM
0  04E3 D312                OUT  PC
0  04E5 DB10                IN   PA
0  04E7 D601                SUI  01
0  04E9 CDF204              CALL PRAD  ; PRINT THE VOLTAGE
0  04EC CD0C09              CALL crlf2
0  04EF C30004              JMP  TR
1                ;
1                ;
0  04F2 FEC8         PRAD:  CPI  0C8H ; PRINT THE ACCUMULATOR IN DECIMAL
0  04F4 DA0105              JC   PRD1
0  04F7 D6C8                SUI  0C8H
0  04F9 0E32                MVI  C,32H
0  04FB CD2C05              CALL COS
0  04FE C30D05              JMP  PRD2
0  0501 FE64         PRD1:  CPI  64H  ; NUMBER LESS THAN 200
0  0503 DA0D05              JC   PRD2 ; LESS THAN 100
0  0506 D664                SUI  64H
0  0508 0E31                MVI  C,31H
0  050A CD2C05              CALL COS
```

```
0  050D 0E30          PRD2: MVI  C,30H
0  050F FE0A          PRD3: CPI  0aH
0  0511 D21A05              JNC  PRD4 ;  LESS THAN 9
0  0514 CD2C05              CALL COS
0  0517 C32005              JMP  PRD5
0  051A D60A          PRD4: SUI  0AH
0  051C 0C                  INR  C
0  051D C30F05
JMP  PRD3 ; KEEP SUBTRACTING 10 UNTIL LESS THAN 10
0  0520 0E2E          PRD5: MVI  C,2EH
0  0522 CD2C05              CALL COS  ;  SEND A DECIMAL POINT
0  0525 C630                ADI  30H
0  0527 4F                  MOV  C,A
0  0528 CD1709              CALL c0
0  052B C9                  RET
0  052C F5            COS: PUSH PSW  ; SAVE ACC.
0  052D CD1709              CALL c0
0  0530 F1                  POP PSW
0  0531 C9                  RET
0  0532 1609          PBIN: MVI  D,09H
0  0534 15            PBN1: DCR  D    ;  SEND OUT ACC. IN BINARY
0  0535 C8                  RZ  ;lets try this
0  0536 B7                  ORA  A
0  0537 17                  RAL
0  0538 DA4305              JC   PBN2
0  053B 0E30                MVI  C,30H
0  053D CD2C05              CALL COS
0  0540 C33405              JMP  PBN1
0  0543 0E31          PBN2: MVI  C,31H
0  0545 CD2C05              CALL COS
0  0548 C33405              JMP  PBN1
1                  ;
1                  ;
0  054B 16FF          TBIN: MVI  D,0FFH
0  054D 7C
MOV  A,H  ;   BINARY TRANSMIT,SET D COUNTER FOR 256 CHAR
0  054E 320E40              STA  PSV3 ;   TRANSMIT LIMIT
0  0551 7D                  MOV  A,L
0  0552 320F40
STA  PSV4 ;   SAVE ADDRESS FOR REPEAT IF BAD TRANSMISSION
0  0555 3E00                MVI  A,00
0  0557 321540              STA  BNFL  ;   WHAT IS THIS FOR?
0  055A 7E            TBN1: MOV  A,M   ; GET A CHAR FROM STORE
0  055B FE0E                CPI  0EH   ;END OF STORE MARKER
0  055D C28905              JNZ  TBN2
0  0560 CD6507
CALL INXP  ;  BUMP THE POINTER ON THE CIRCULAR STORE
0  0563 7E                  MOV  A,M
0  0564 FE0E                CPI  0EH
0  0566 C28505
JNZ  CON1  ; TO DECREASE THE POINTER TO THE RIGHT PLACE AGAIN
0  0569 CD6507              CALL INXP
0  056C 7E                  MOV  A,M
0  056D FE0E                CPI  0EH
0  056F C28205              JNZ  CON2
0  0572 CD6E07              CALL DCXP
```

```
0  0575 CD6E07              CALL DCXP  ; SET THE POINTER TO THE END AGAIN
0  0578 F5                  PUSH PSW
0  0579 3E01                MVI  A,01H
0  057B 321540              STA  BNFL
0  057E F1                  POP  PSW
0  057F C39305              JMP  TB2   ;  SET BNFL TO SHOW END OF STORE SENT
0  0582 CD6E07       CON2:  CALL DCXP
0  0585 CD6E07       CON1:  CALL DCXP  ;  RESET POINTER
0  0588 7E                  MOV  A,M
0  0589 CD4107       TBN2:  CALL PRAC  ;  SEND THE CHAR
0  058C CD6507              CALL INXP
0  058F 15                  DCR  D    ;  ENOUGH SENT?
0  0590 C25A05              JNZ  TBN1
0  0593 E5           TB2:   PUSH H    ;   SAVE THE POINTER
0  0594 CD0E0D              CALL CKSC ;  ANY SENSORS?
0  0597 E1                  POP  H
0  0598 0E3F                MVI  C,3FH
0  059A CD1709              CALL c0 ;  SEND A'?'.
0  059D CDC307              CALL ci2
0  05A0 FE0D                CPI  0DH  ;  IS CR?
0  05A2 CAB505              JZ   TBN3 ;  IF SO,REPEAT THE LAST BLOCK.
0  05A5 FE1B                CPI  1BH  ;  ESCAPE?
0  05A7 CAC005              JZ   TBN4 ;  GIVE UP TRANSMISSON
0  05AA 3A1540
LDA  BNFL ;  ANYTHING ELSE,CHECK IF END OF DATA,IF NOT,CONTINUE
0  05AD FE01                CPI  01H
0  05AF CAC005              JZ   TBN4
0  05B2 C34B05              JMP  TBIN
0  05B5 3A0E40       TBN3:  LDA PSV3
0  05B8 67                  MOV  H,A
0  05B9 3A0F40              LDA  PSV4
0  05BC 6F                  MOV  L,A
0  05BD C34B05              JMP TBIN
0  05C0 0E04         TBN4:  MVI  C,04H  ; SEND OUT EOF,THEN RETURN.
0  05C2 CD1709              CALL c0
0  05C5 C9                  RET
0  05C6 110050
FMDN: LXI  D, 5000H   ; SET UPPER LIMIT ON CHARS TO CHECK
0  05C9 1B           FMD1:  DCX  D
0  05CA 3E00                MVI  A,00
0  05CC BA                  CMP  D
0  05CD C8                  RZ
0  05CE CD6E07              CALL DCXP
0  05D1 3EB3                MVI  A,0B3H
0  05D3 BE                  CMP  M ;   LOOK FOR SEQUENCE 'B3 xx xx xx A3'
0  05D4 C2C905              JNZ  FMD1 ;  TO SHOW START OF DATE FIELD
0  05D7 E5                  PUSH H
0  05D8 CD6507              CALL INXP
0  05DB CD6507              CALL INXP
0  05DE CD6507              CALL INXP
0  05E1 3EA3                MVI  A,0A3H
0  05E3 BE                  CMP  M
0  05E4 CAEB05              JZ   FMD2
0  05E7 E1                  POP  H
0  05E8 C3C905              JMP  FMD1 ;  NO SUCCESS,SO TRY AGAIN
0  05EB CD6E07       FMD2:  CALL DCXP
```

```
0  05EE CD6E07              CALL DCXP ;  RESET THE DATA POINTER
0  05F1 CD6E07              CALL DCXP
0  05F4 C601                ADI  01
0  05F6 33                  INX  SP
0  05F7 33                  INX  SP   ;  CORRECT THE STACK POINTER
0  05F8 C9                  RET
0  05F9 CD6507       TAS1:  CALL INXP  ; ROUTINE TO SEND BACK DATA IN ASCII
0  05FC DBED         TASC:  IN   UC    ; THIS LOOKS FOR ESCAPE,+ OR-
0  05FE E602                ANI  02    ; AND REACTS ACCORDINGLY
0  0600 CA1406              JZ   TAS2
0  0603 DBEC                IN   UD
0  0605 E67F                ANI  7FH
0  0607 FE1B                CPI  1BH  ; ESC
0  0609 C8                  RZ         ;  GO BACK TO TR
0  060A FE2B                CPI  2BH
0  060C CCA207              CZ   IXPT  ; IF PLUS,UP THE POINTER
0  060F FE2D                CPI  2DH
0  0611 CCAE07              CZ   DXPT
0  0614 7E
TAS2: MOV  A,M  ; LOOK AT THE DATA AND DECIDE WHAT TO DO
0  0615 FE0E                CPI  0EH  ;  EOF
0  0617 C8                  RZ
0  0618 FE0F                CPI  0FH  ;  EOL
0  061A CA7306              JZ   CRL1
0  061D FEB3                CPI  0B3H  ; DATE
0  061F CA7906              JZ   B3
0  0622 FEC3                CPI  0C3H  ; VISIT
0  0624 CA9E06              JZ   C3
0  0627 FEA2                CPI  0A2H ;  SENSORS
0  0629 CAB606              JZ   A2
0  062C FEA3                CPI  0A3H  ;  TIME
0  062E CAE906              JZ   A3
0  0631 FEF2                CPI  0F2H  ;  HEIGHT
0  0633 CAEF06              JZ   F2
0  0636 FED3                CPI  0D3H  ;  DATUM
0  0638 CA8307              JZ   D3
0  063B 16FF
ERCH: MVI  D,0FFH  ;  SEGMENT TO CORRECT ERRORS IF NOT A RECOGNIZED CODE
0  063D 15           ERC1:  DCR  D        ;  LIMIT THE NO. OF TRIES
0  063E CA6A06              JZ   MJER     ;  A MAJOR ERROR
0  0641 CD6507              CALL INXP
0  0644 3EF2                MVI  A,0F2H
0  0646 BE
CMP  M        ;  LOOK FOR TWO OCCURENCES OF F2 xx xx 0F
0  0647 C23D06              JNZ  ERC1
0  064A CD6507              CALL INXP
0  064D CD6507              CALL INXP
0  0650 CD6507              CALL INXP
0  0653 3E0F                MVI  A,0FH
0  0655 BE                  CMP  M
0  0656 CA6406              JZ   EROK    ;  ERROR RECOVERED
0  0659 CD6E07              CALL DCXP
0  065C CD6E07              CALL DCXP    ; RESET THE POINTER
0  065F 3EF2                MVI  A,0F2H
0  0661 C33D06              JMP  ERC1
0  0664 CD6507
```

```
EROK: CALL INXP   ; WRAP THINGS UP AND GO BACK TO INTERPRET THE LINE
0  0667 C3FC05              JMP  TASC
0  066A 0E21          MJER: MVI  C,21H  ; LET THEM KNOW YOU CAN'T RECOVER
0  066C CD1709              CALL c0
0  066F CD0C09              CALL crlf2
0  0672 C9                  RET
0  0673 CD0C09        CRL1: CALL crlf2
0  0676 C3F905
JMP  TAS1  ;  RESET THE CARRIAGE \001AND THEN BACK TO PROG
0  0679 CD0C09        B3: CALL crlf2  ; PRINT THE DATE
0  067C E5                  PUSH H   ; PRESERVE THE DATA POINTER
0  067D CD2F07              CALL APRT
0  0680 4441544520   DB    'DATE '
0  0681 04                  DB   04H
0  0686 E1                  POP  H
0  0687 CD6507              CALL INXP
0  068A 7E                  MOV  A,M
0  068B CD4107              CALL PRAC
0  068E CD6507              CALL INXP
0  0691 CD2907              CALL SPCE
0  0694 7E                  MOV  A,M
0  0695 CD4107              CALL PRAC
0  0698 CD0C09              CALL  crlf2
0  069B C3F905              JMP  TAS1
0  069E CD0C09        C3: CALL crlf2
0  06A1 E5                  PUSH H
0  06A2 CD2F07              CALL  APRT
0  06A5 444154554D   DB  'DATUM '
0       20
0  06A6 04                  DB  04H
0  06AC E1                  POP  H
0  06AD CD0307              CALL TPRT
0  06B0 CD0C09              CALL crlf2
0  06B3 C3F905              JMP  TAS1
0  06B6 CD0C09        A2: CALL crlf2
0  06B9 E5                  PUSH H
0  06BA CD2F07              CALL APRT
0  06BD 53454E534F   DB    'SENSORS '
0       525320
0  06BE 04                  DB    04H
0  06C6 E1                  POP   H
0  06C7 CD6507        A21: CALL INXP
0  06CA 7E                  MOV  A,M
0  06CB CD4107              CALL PRAC
0  06CE CD6507              CALL INXP
0  06D1 7E                  MOV  A,M
0  06D2 CD4107              CALL PRAC
0  06D5 CD6507              CALL INXP
0  06D8 0E20                MVI  C,20H
0  06DA CD1709              CALL c0
0  06DD 7E                  MOV  A,M
0  06DE FEA2                CPI  0A2H
0  06E0 CAC706              JZ   A21
0  06E3 CD0C09              CALL crlf2
0  06E6 C3FC05              JMP  TASC
0  06E9 CD0307        A3: CALL TPRT  ; PRINT THE TIME
```

```
0  06EC C3F905              JMP  TAS1
0  06EF CD6507       F2: CALL INXP  ; PRINT THE HEIGHT FROM A SENSOR
0  06F2 56                  MOV  D,M
0  06F3 CD6507              CALL INXP
0  06F6 5E                  MOV  E,M  ; PUT THE READING IN DE PAIR
0  06F7 E5                  PUSH H
0  06F8 EB                  XCHG
0  06F9 CDB50A              CALL SPRT  ; GO PRINT THE SIGNED HEIGHT
0  06FC E1                  POP  H
0  06FD CD2907              CALL SPCE
0  0700 C3F905              JMP  TAS1
0  0703 CD6507       TPRT: CALL INXP  ; PRINT THE TIME
0  0706 7E                  MOV  A,M
0  0707 CD4107              CALL PRAC
0  070A 0E3A                MVI  C,':'
0  070C CD1709              CALL c0
0  070F CD6507              CALL INXP
0  0712 7E                  MOV  A,M
0  0713 CD4107              CALL PRAC
0  0716 0E3A                MVI  C,':'
0  0718 CD1709              CALL c0
0  071B CD6507              CALL INXP
0  071E 7E                  MOV  A,M
0  071F CD4107              CALL PRAC
0  0722 CD2907              CALL SPCE
0  0725 CD2907              CALL SPCE
0  0728 C9                  RET
0  0729 0E20         SPCE: MVI  C,' '  ; PRINT A SPACE
0  072B CD1709              CALL c0
0  072E C9                  RET
0  072F E3           APRT: XTHL  ; PRINT AN ASCII STRING FROM THE PROG
0  0730 4E
SAPT: MOV  C,M ;STARTING AT THE RETURN ADDR. ON THE STACK
0  0731 3E04                MVI  A,04H  ; THIS EOF DENOTES THE END OF STRING
0  0733 B9                  CMP  C
0  0734 CA3E07              JZ   RAPT  ; WRAP-UP ROUTINE
0  0737 CD1709              CALL c0
0  073A 23                  INX  H
0  073B C33007              JMP SAPT
0  073E 23
RAPT: INX  H  ; RESTORE THE STACK TOP FOR CORRECT RETURN
0  073F E3                  XTHL   ;
0  0740 C9                  RET  ;
0  0741 F5           PRAC: PUSH PSW  ; PRINT THE ACC. AS TWO ASCII BYTES
0  0742 0F                  RRC
0  0743 0F                  RRC     ; FIRST GET THE HIGH NIBBLE
0  0744 0F                  RRC
0  0745 0F                  RRC   ;
0  0746 E60F                ANI  0FH  ;GET RID OF THE REMNANTS
0  0748 CD5207              CALL PUTH  ; SEND AS ASCII
0  074B F1                  POP  PSW
0  074C E60F                ANI  0FH  ; NOW THE LOW NIBBLE
0  074E CD5207              CALL PUTH
0  0751 C9                  RET
0  0752 FE0A         PUTH: CPI  0AH  ; LESS THAN 10?
0  0754 FA5E07              JM   PUTD
```

```
0  0757 C637              ADI  37H  ;GREATER,SO SEND AS LETTER
0  0759 4F                MOV  C,A
0  075A CD1709            CALL cO
0  075D C9                RET
0  075E C630        PUTD: ADI  30H  ; CONVERT TO ASCII NUMBER
0  0760 4F                MOV  C,A
0  0761 CD1709            CALL cO
0  0764 C9                RET
0  0765 23          INXP: INX  H  ;  INCREMENT THE DATA STORE POINTER
0  0766 3E80              MVI  A,80h
0  0768 BC                CMP  H  ; IS AT TOP OF STORE?
0  0769 CO                RNZ
0  076A 210041            LXI  H, 4100h  ; YES,SO RESET TO BOTTOM AGAIN
0  076D C9                RET
0  076E 2B          DCXP: DCX  H  ; DECREMENT THE POINTER
0  076F 3E40              MVI  A,40H
0  0771 BC                CMP  H  ; BOTTOM OF STORE?
0  0772 CO                RNZ
0  0773 21FF7F            LXI  H, 7fffh
0  0776 C9                RET
0  0777 DBED        ci3: IN   UC  ; GET A CHAR FROM THE USART
0  0779 E602              ANI  02H  ; WAIT FOR RX READY
0  077B CA7707            JZ   ci3
0  077E DBEC              IN   UD  ;GET THE CHAR
0  0780 E67F              ANI  7FH  ;GET RID OF PARITY
0  0782 C9                RET  ;
0  0783 CD0C09      D3: CALL crlf2  ;  PRINT THE DATUM HEIGHT
0  0786 E5                PUSH H
0  0787 CD2F07            CALL APRT
0  078A 444154554D  DB   'DATUM  '
0       2020
0  078B 04                DB   04H
0  0792 E1                POP H
0  0793 CD0307            CALL TPRT
0  0796 CD0307            CALL TPRT
0  0799 CD0307            CALL TPRT
0  079C CD0C09            CALL crlf2
0  079F C3F905            JMP  TAS1
0  07A2 3EFF        IXPT: MVI  A,0FFH  ;INCREASE THE POINTER BY 256
0  07A4 F5          IXP1: PUSH PSW
0  07A5 CD6507            CALL INXP
0  07A8 F1                POP  PSW
0  07A9 3D                DCR  A
0  07AA C8                RZ
0  07AB C3A407            JMP IXP1
0  07AE 3EFF        DXPT: MVI  A,0FFH
0  07B0 F5          DXP1: PUSH PSW
0  07B1 CD6E07            CALL DCXP
0  07B4 F1                POP  PSW
0  07B5 3D                DCR  A
0  07B6 C8                RZ  ;and again
0  07B7 C3B007            JMP  DXP1
0  07BA 67          ltr:  mov  h,a
0  07BB 2E00              mvi  l,0
0  07BD CD4B05            call  tbin
0  07C0 C30004            jmp  tr
```

```
0  07C3 DBEC        ci2:  in  ud
0  07C5 CD7707             call  ci3
0  07C8 FE1B               cpi  esc
0  07CA C2C307             jnz  ci2
0  07CD CD7707             call  ci3
0  07D0 FE1B               cpi  esc
0  07D2 C8                 rz
0  07D3 FE20               cpi  sp
0  07D5 C8                 rz
0  07D6 FE0D               cpi  cr
0  07D8 C8                 rz
0  07D9 C3C307             jmp  ci2
o1                 org 0800h
0  0800 CD0C09     PMPT:   CALL  crlf2
0  0803 0E2A               MVI   C,'*'  ; SEND A STAR TO TERMINAL
0  0805 CD1709             CALL  c0
0  0808 CDBA08             CALL  biln    ;  GET SOME RESPONSE
0  080B 21C03D             LXI   H,3dc0h
0  080E 56                 MOV   D,M     ;  PUT TWO CHAR IN DE
0  080F 23                 INX   H
0  0810 5E                 MOV   E,M
0  0811 214E08     COMD:   LXI   H,CMST  ;  FIND OUT WHAT TO DO
0  0814 7E         COMS:   MOV   A,M
0  0815 FE00               CPI   00      ; IS IT A BAD COMMAND?
0  0817 CA4008             JZ    COM4
0  081A FE01               CPI   01      ; OK?
0  081C C22908             JNZ   COMT
0  081F 3AA03E             LDA   EXCF
0  0822 FE01               CPI   01
0  0824 C24008             JNZ   COM4
0  0827 23                 INX   H
0  0828 7E                 MOV   A,M
0  0829 BA         COMT:   CMP   D       ;   SET UP ADDRESS TO EXECUTE
0  082A C23908             JNZ   COM1    ;   COMMAND
0  082D 23                 INX    H
0  082E 7E                 MOV   A,M
0  082F BB                 CMP   E
0  0830 C23A08             JNZ   COM2    ;  IF NOT RECOGNISED,TRY AGAIN
0  0833 23                 INX   H     ;  IS OK!
0  0834 56                 MOV   D,M
0  0835 23                 INX   H
0  0836 5E                 MOV   E,M     ;  PUT THE ADDRESS IN DE
0  0837 EB                 XCHG          ; THEN HL
0  0838 E9                 PCHL          ; GO TO IT
0  0839 23         COM1:   INX   H
0  083A 23         COM2:   INX   H     ; INCREMENT THE POINTER THE
0  083B 23                 INX   H     ; RIGHT NO. OF TIMES
0  083C 23                 INX   H
0  083D C31408             JMP   COMS    ;  THEN TRY AGAIN
0  0840 0E3F       COM4:   MVI   C,'?'   ;  DOES NOT RECOGNISE COMMAND
0  0842 CD1709             CALL  c0
0  0845 CD0C09             CALL  crlf2   ; TELL THE TERMINAL
0  0848 C30008             JMP   PMPT
0  084B C39A09     RCL1:   JMP   RCLK    ;  ?????
1              ;
1              ;  NOW PUT IN THE LIST OF COMMANDS AND ADDRESSES
```

```
1                       ;
0   084E 5254       cmst: DB    'RT'
0   0850 088D             dwr   RT
0   0852 5244       DB    'RD'
0   0854 08AC             dwr   RD
0   0856 5452       DB    'TR'
0   0858 0400             dwr   TR
0   085A 454E       DB    'EN'
0   085C 087A             dwr   RET
0   085D 01               DB    01H
0   085F 5354       DB    'ST'
0   0861 087B             dwr   ST
0   0863 5344       DB    'SD'
0   0865 0884             dwr   SD
0   0867 5349       DB    'SI'
0   0869 0B08             dwr   SI
0   086B 5350       DB    'SP'
0   086D 0B3D             dwr   SPTR
0   086F 494E       DB    'IN'
0   0871 0B1C             dwr   INIT
0   0873 4C50       DB    'LP'
0   0875 0B4C             dwr   LPTR
0   0876 00               DB    00H
0   0878 0800             dwr   PMPT
0   087A C9         RET:  RET
0   087B 010407     ST:   LXI   B,0704H
0   087E CD4F09           CALL  SCLK
0   0881 C30008           JMP   PMPT
0   0884 01060D     SD:   LXI   B,0D06H
0   0887 CD4F09           CALL  SCLK
0   088A C30008           JMP   PMPT
0   088D 010607     RT:   LXI   B,0706H
0   0890 CD9A09           CALL  RCLK
0   0893 0E07             MVI   C,07H
0   0895 CD1709           CALL  c0
0   0898 21903D           LXI   H,3D90H
0   089B 1606             MVI   D,06
0   089D 7E         RCFT: MOV   A,M
0   089E C630             ADI   30H
0   08A0 4F               MOV   C,A
0   08A1 CD1709           CALL  c0
0   08A4 23               INX   H
0   08A5 15               DCR   D
0   08A6 C29D08           JNZ   RCFT
0   08A9 C30008           JMP   PMPT
0   08AC 01050C     RD:   LXI   B,0C05H
0   08AF CD9A09           CALL  RCLK
0   08B2 21903D           LXI   H,3D90H
0   08B5 1605             MVI   D,05H
0   08B7 C39D08           JMP   RCFT
1                 ;  biln is the routine to build an input line from the
1                 ;  rs232 interface.
0   08BA 3E28       biln:  mvi  a,28h   ;first reset the timer
0   08BC D312              out  pc
0   08BE 3E20              mvi  a,20h
0   08C0 D312              out  pc
```

```
0  08C2 21C03D      biln1: lxi  h,3dc0h  ;this is where to write to
0  08C5 DB10      biln2: in   pa       ;check the timer
0  08C7 E680             ani  80h
0  08C9 FE80             cpi  80h
0  08CB CC0E0D           cz   cksc     ;check for a scan if timed out
0  08CE DBED             in   uc       ;now check the USART
0  08D0 E602             ani  02
0  08D2 CAC508           jz   biln2
0  08D5 DBEC             in   ud       ;read the char
0  08D7 E67F             ani  7fh
0  08D9 FE7F             cpi  7fh      ;clear parity,check delete
0  08DB C2ED08           jnz  biln3
0  08DE 3EC0             mvi  a,0c0h   ;is delete
0  08E0 BD               cmp  l      ;start of line?
0  08E1 CAC508           jz   biln2
0  08E4 0E2A             mvi  c,2ah
0  08E6 CD1709           call c0
0  08E9 2D               dcr  l
0  08EA C3C508           jmp  biln2
0  08ED 77        biln3: mov  m,a    ;store the char
0  08EE 2C               inr  l
0  08EF CA0109           jz   error  ;is line to long?
0  08F2 FE0D             cpi  0dh
0  08F4 C8               rz
0  08F5 FE1B             cpi  1bh
0  08F7 CA0109           jz  error
0  08FA 4F               mov  c,a
0  08FB CD1709           call  c0   ;ech0 the char
0  08FE C3C508           jmp  biln2
0  0901 0E21      error: mvi  c,21h
0  0903 CD1709           call c0
0  0906 CD0C09           call crlf2  ;send out !
0  0909 C3C208           jmp  biln1
0  090C 0E0A      crlf2: mvi  c,0ah
0  090E CD1709           call c0    ;send cr/lf
0  0911 0E0D             mvi  c,0dh
0  0913 CD1709           call c0
0  0916 C9               ret
0  0917 DBED      c0:    in   uc
0  0919 E601             ani  01
0  091B CA1709           jz   c0
0  091E 79               mov  a,c
0  091F D3EC             out  ud    ;send out char from c reg
0  0921 C9               ret
1                ;get the parameters from the command line
1                ;  C  reg contains no. of params to get
0  0922 21C03D     gtpm:  lxi  h,3dc0h  ;point at the line
0  0925 11A03D            lxi  d,3da0h  ;and where to put the parameters
0  0928 0620              mvi  b,' '    ;to look for a space
0  092A 7E         gtp1:  mov  a,m
0  092B 23                inx  h
0  092C B8                cmp  b
0  092D C22A09            jnz  gtp1  ;look for a space
0  0930 7E         gtp2:  mov  a,m
0  0931 23                inx  h
0  0932 B8                cmp  b
```

```
0  0933 CA3009              jz  gtp2  ;now for none-blank
0  0936 C6D0                adi   0d0h
0  0938 D24909              jnc  ergt  ;less than 30h
0  093B C6F6                adi  0f6h
0  093D DA4909              jc   ergt  ;>9
0  0940 C60A                adi  10    ;make it numeric
0  0942 12                  stax  d
0  0943 13                  inx  d
0  0944 0D                  dcr  c     ;is that all wanted?
0  0945 C23009              jnz  gtp2
0  0948 C9                  ret
0  0949 0E21         ergt:  mvi  c,'!'
0  094B CD1709              call c0
0  094E C9                  ret
1                  ;  segment to set the clock
0  094F C5           sclk:  push  b
0  0950 CD2209              call gtpm
0  0953 3E21                mvi  a,21h
0  0955 B9                  cmp  c
0  0956 C1                  pop  b
0  0957 C8                  rz
0  0958 21A03D              lxi   h,3da0h
0  095B AF                  xra  a
0  095C CD8709              call  wclk
0  095F 3E0E                mvi   a,0eh
0  0961 CD8709              call wclk   ;  stop the clock
0  0964 56           stst:  mov  d,m    ;  transfer the data into the clock
0  0965 CD8109              call wcdb
0  0968 23                  inx  h
0  0969 05                  dcr  b
0  096A C3D00B              jmp st2
0  096D 0E07         ret1:  mvi  c,7
0  096F CD1709              call  c0  ;send out a bell,then wait
0  0972 DBED         wait:  in   uc
0  0974 E602                ani  02
0  0976 CA7209              jz  wait
0  0979 DBEC                in  ud
0  097B 3E1E                mvi  a,1eh
0  097D CD8709              call  wclk
0  0980 C9                  ret
0  0981 7A           wcdb:  mov  a,d  ;put out data to clk.
0  0982 07                  rlc
0  0983 07                  rlc
0  0984 07                  rlc
0  0985 07                  rlc
0  0986 80                  add  b
0  0987 D311         wclk:  out  pb
0  0989 3E19                mvi  a,19h   ;this sends a write strobe
0  098B D312                out  pc
0  098D 3E13                mvi  a,13h
0  098F D312                out  pc
0  0991 3E17                mvi  a,17h
0  0993 D312                out  pc
0  0995 3E13                mvi  a,13h
0  0997 D312                out  pc
0  0999 C9                  ret
```

```
1                ;   this is the clock read routine.
1                ;
1                ;
0  099A D5           rclk:  push  d
0  099B 16FF              mvi  d,0ffh
0  099D 219D3D       rcl:   lxi  h,3d90h
0  09A0 C5                  push  b
0  09A1 78           lcst:  mov  a,b
0  09A2 D311                out  pb
0  09A4 3E10         lcs1: mvi  a,10h
0  09A6 D312                out  pc
0  09A8 DB10                in  pa
0  09AA 77                  mov  m,a
0  09AB AF                  xra  a
0  09AC D312                out  pc
0  09AE 7E                  mov  a,m
0  09AF E60F                ani  0fh
0  09B1 FE0F                cpi  0fh  ;is it error?
0  09B3 C2BD09              jnz  lcs2
0  09B6 C1                  pop  b
0  09B7 15                  dcr  d
0  09B8 C29D09              jnz   rcl
0  09BB D1                  pop   d
0  09BC C9                  ret
0  09BD 23           lcs2:  inx  h
0  09BE 05                  dcr  b
0  09BF 0D                  dcr  c
0  09C0 C2A109              jnz   lcst
0  09C3 C1                  pop  b
0  09C4 D1                  pop  d
0  09C5 C9                  ret
1                ;
1                ;
1                ;
1                ;  THIS SEGMENT looks after reading the
1                ;  sensor card,and translating the
1                ;  reading into a height by doing the
1                ;  quadratic conversion.
1                ;
0  09C6 07            gtht:  rlc
0  09C7 07                   rlc
0  09C8 07                   rlc
0  09C9 07                   rlc  ;  put the slot no. into
0  09CA F5                   push psw;   high nibble and save
0  09CB C602                 adi  02
0  09CD D312                 out  pc
0  09CF 3E00                 mvi  a,0
0  09D1 D311                 out  pb
0  09D3 DB10                 in  pa
0  09D5 FEAA                 cpi  0aah  ;  is a sensor present?
0  09D7 CADD09               jz  gth1
0  09DA F1                   pop  psw
0  09DB 37                   stc
0  09DC C9                   ret   ; if not,set carry & return
0  09DD 0602         gth1:   mvi  b,02
0  09DF 0E06                 mvi  c,06
```

```
0  09E1 21023F            lxi  h,3f02h
0  09E4 78          gth2: mov  a,b
0  09E5 D311              out  pb
0  09E7 DB10              in  pa
0  09E9 77                mov  m,a
0  09EA 04                inr  b
0  09EB 23                inx  h
0  09EC 0D                dcr  c
0  09ED C2E409            jnz  gth2  ;get the coeffs
0  09F0 F1                pop  psw
0  09F1 F5                push  psw  ;restore the slot
0  09F2 21013F            lxi  h,3f01h
0  09F5 D312              out  pc
0  09F7 DB10              in   pa
0  09F9 77                mov  m,a
0  09FA 2B                dcx  h
0  09FB F1                pop  psw
0  09FC C601              adi  01
0  09FE D312              out  pc
0  0A00 DB10              in  pa
0  0A02 77                mov  m,a
0  0A03 CD080A            call  quad
0  0A06 B7                ora  a
0  0A07 C9                ret
0  0A08 21023F      quad: lxi  h,3f02h
0  0A0B 56                mov  d,m
0  0A0C 2C                inr  l
0  0A0D 5E                mov  e,m
0  0A0E D5                push  d
0  0A0F 21043F            lxi  h,3f04h
0  0A12 46                mov  b,m
0  0A13 2C                inr  l
0  0A14 4E                mov  c,m
0  0A15 21003F            lxi  h,3f00h
0  0A18 56                mov  d,m
0  0A19 2C                inr  l
0  0A1A 5E                mov  e,m
0  0A1B 78                mov  a,b
0  0A1C E680              ani  80h
0  0A1E 32083F            sta  sflg
0  0A21 78                mov   a,b
0  0A22 E67F              ani  7fh
0  0A24 47                mov  b,a
0  0A25 CD5C0A            call  mult
0  0A28 3A083F            lda  sflg
0  0A2B B4                ora  h
0  0A2C 67                mov  h,a
0  0A2D D1                pop  d
0  0A2E CD7E0A            call  add
0  0A31 E5                push  h
0  0A32 21003F            lxi  h,3f00h
0  0A35 46                mov  b,m
0  0A36 2C                inr  l
0  0A37 4E                mov  c,m
0  0A38 50                mov  d,b
0  0A39 59                mov  e,c
```

```
0  0A3A CD5C0A            call  mult
0  0A3D E5                push  h
0  0A3E 21063F            lxi   h,3f06h
0  0A41 46                mov   b,m
0  0A42 2C                inr   l
0  0A43 4E                mov   c,m
0  0A44 D1                pop   d
0  0A45 78                mov   a,b
0  0A46 E680              ani   80h
0  0A48 32083F            sta   sflg
0  0A4B 78                mov   a,b
0  0A4C E67F              ani   7fh
0  0A4E 47                mov   b,a
0  0A4F CD5C0A            call  mult
0  0A52 3A083F            lda   sflg
0  0A55 B4                ora   h
0  0A56 67                mov   h,a
0  0A57 D1                pop   d
0  0A58 CD7E0A            call  add
0  0A5B C9                ret
0  0A5C 210000      mult: lxi   h,0
0  0A5F AF          strt: xra   a
0  0A60 7A                mov   a,d
0  0A61 1F                rar
0  0A62 57                mov   d,a
0  0A63 7B                mov   a,e
0  0A64 1F                rar
0  0A65 5F                mov   e,a
0  0A66 AF                xra   a
0  0A67 79                mov   a,c
0  0A68 17                ral
0  0A69 4F                mov   c,a
0  0A6A 78                mov   a,b
0  0A6B 17                ral
0  0A6C 47                mov   b,a
0  0A6D D2740A            jnc   chek
0  0A70 19                dad   d
0  0A71 C35F0A            jmp   strt
0  0A74 AF          chek: xra   a
0  0A75 B8                cmp   b
0  0A76 C25F0A            jnz strt
0  0A79 B9                cmp   c
0  0A7A C25F0A            jnz   strt
0  0A7D C9                ret
0  0A7E 7C          add:  mov   a,h
0  0A7F E680               ani   80h
0  0A81 FE80               cpi   80h
0  0A83 C2910A             jnz   sad1
0  0A86 7C                 mov   a,h
0  0A87 E67F               ani   7fh
0  0A89 67                 mov   h,a
0  0A8A AF                 xra   a
0  0A8B 95                 sub   l
0  0A8C 6F                 mov   l,a
0  0A8D 3E00               mvi   a,0
0  0A8F 9C                 sbb   h
```

```
0  0A90 67                    mov  h,a
0  0A91 7A           sad1:    mov  a,d
0  0A92 E680                  ani  80h
0  0A94 FE80                  cpi  80h
0  0A96 C2A40A                jnz  sad2
0  0A99 7A                    mov  a,d
0  0A9A E67F                  ani  7fh
0  0A9C 57                    mov  d,a
0  0A9D AF                    xra  a
0  0A9E 93                    sub  e
0  0A9F 5F                    mov  e,a
0  0AA0 3E00                  mvi  a,0
0  0AA2 9A                    sbb  d
0  0AA3 57                    mov  d,a
0  0AA4 19           sad2:    dad  d
0  0AA5 7C                    mov  a,h
0  0AA6 E680                  ani  80h
0  0AA8 FE80                  cpi  80h
0  0AAA C0                    rnz
0  0AAB AF                    xra  a
0  0AAC 95                    sub  l
0  0AAD 6F                    mov  l,a
0  0AAE 3E00                  mvi  a,0
0  0AB0 9C                    sbb  h
0  0AB1 F680                  ori  80h
0  0AB3 67                    mov  h,a
0  0AB4 C9                    ret
1                   SFLG:  equ  3f08h
0  0AB5 7C             sprt:  mov  a,h    ;  print signed no.
0  0AB6 C680                  adi  80h
0  0AB8 DAC30A                jc   mnus
0  0ABB 0E2B                  mvi  c,'+'
0  0ABD CD1709                call  c0
0  0AC0 C3CC0A                jmp prt
0  0AC3 0E2D           mnus:  mvi  c,'-'
0  0AC5 CD1709                call  c0
0  0AC8 7C                    mov  a,h
0  0AC9 E67F                  ani  7fh
0  0ACB 67                    mov  h,a
0  0ACC 111027         prt:  lxi  d,10000
0  0ACF CDEB0A                call sub
0  0AD2 11E803                lxi  d,1000
0  0AD5 CDEB0A                call  sub
0  0AD8 116400                lxi  d,100
0  0ADB CDEB0A                call  sub
0  0ADE 110A00                lxi  d,10     lxi  d,10
0  0AE1 CDEB0A                call  sub
0  0AE4 110100                lxi  d,1
0  0AE7 CDEB0A                call sub
0  0AEA C9                    ret
0  0AEB 0E30           sub:  mvi  c,'0'
0  0AED 7C             ne:    mov  a,h
0  0AEE BA                    cmp  d
0  0AEF DA040B                jc  pri
0  0AF2 C2FA0A                jnz  inc
0  0AF5 7D                    mov  a,l
```

```
0  0AF6 BB                  cmp  e
0  0AF7 DA040B              jc   pri
0  0AFA 7D           inc:   mov  a,l
0  0AFB 93                  sub  e
0  0AFC 6F                  mov  l,a
0  0AFD 7C                  mov  a,h
0  0AFE 9A                  sbb  d
0  0AFF 67                  mov  h,a
0  0B00 0C                  inr  c
0  0B01 C3ED0A              jmp  ne
0  0B04 CD1709       pri:   call c0
0  0B07 C9                  ret
1                    ;  Command execution block.
0  0B08 0E01         si:    mvi  c,1    ;set int. period
0  0B0A CD2209              call gtpm
0  0B0D 3AA03D              lda  3da0h
0  0B10 D311                out  pb
0  0B12 3E1C                mvi  a,1ch
0  0B14 D312                out  pc
0  0B16 AF                  xra  a
0  0B17 D312                out  pc
0  0B19 C30008              jmp  pmpt
0  0B1C 3E41         init:  mvi  a,41h  ;start the thing up
0  0B1E 321140              sta  ptr1
0  0B21 321340              sta  psv1
0  0B24 AF                  xra  a
0  0B25 321740              sta    dmfl
0  0B28 321840              sta  dfl2
0  0B2B 321240              sta  ptr2
0  0B2E 321440              sta  psv2
0  0B31 CD1E0C              call  lgdt
0  0B34 CD260D              call  snlg
0  0B37 CDB90C              call  ceof
0  0B3A C30008              jmp  pmpt
0  0B3D 3A1140       sptr:  lda  ptr1  ;update the tr pointer
0  0B40 321340              sta  psv1
0  0B43 3A1240              lda ptr2
0  0B46 321440              sta  psv2
0  0B49 C30008              jmp  pmpt
0  0B4C AF           lptr:  xra  a
0  0B4D 67                  mov  h,a
0  0B4E 6F                  mov  l,a
0  0B4F 0E40                mvi  c,40h
0  0B51 CD1709              call c0
0  0B54 CDB50B       blh1:  call cin
0  0B57 4F                  mov  c,a
0  0B58 CD1709              call  c0
0  0B5B 79                  mov  a,c
0  0B5C FE1B                cpi  1bh
0  0B5E CA0008              jz  pmpt
0  0B61 FE0D                cpi  0dh
0  0B63 CA8C0B              jz  lpt1
0  0B66 FE30                cpi  '0'
0  0B68 DA840B              jc  bdch
0  0B6B FE3A                cpi  3ah
0  0B6D DA7C0B              jc  chok
```

```
0  0B70 FE41              cpi  'A'
0  0B72 DA840B            jc   bdch
0  0B75 FE47              cpi  'G'
0  0B77 D2840B            jnc  bdch
0  0B7A D607              sui  07
0  0B7C D630       chok:  sui  '0'
0  0B7E CD970B            call rhll
0  0B81 C3540B            jmp  blh1
0  0B84 0E21       bdch:  mvi  c,'!'
0  0B86 CD1709            call c0
0  0B89 C3540B            jmp  blh1
0  0B8C 7C         lpt1:  mov  a,h
0  0B8D 321340            sta  psv1
0  0B90 7D                mov  a,l
0  0B91 321440            sta  psv2
0  0B94 C30008            jmp  pmpt
0  0B97 F5         rhll:  push psw
0  0B98 7C                mov  a,h
0  0B99 E60F              ani  0fh
0  0B9B 17                ral
0  0B9C 17                ral
0  0B9D 17                ral
0  0B9E 17                ral
0  0B9F 67                mov  h,a
0  0BA0 7D                mov  a,l
0  0BA1 E6F0              ani  0f0h
0  0BA3 1F                rar
0  0BA4 1F                rar
0  0BA5 1F                rar
0  0BA6 1F                rar
0  0BA7 84                add  h
0  0BA8 67                mov  h,a
0  0BA9 7D                mov  a,l
0  0BAA E60F              ani  0fh
0  0BAC 17                ral
0  0BAD 17                ral
0  0BAE 17                ral
0  0BAF 17                ral
0  0BB0 6F                mov  l,a
0  0BB1 F1                pop  psw
0  0BB2 85                add  l
0  0BB3 6F                mov  l,a
0  0BB4 C9                ret
0  0BB5 DBED       cin:   in   uc
0  0BB7 E602              ani  02
0  0BB9 CAB50B            jz   cin
0  0BBC DBEC              in   ud
0  0BBE E67F              ani  7fh
0  0BC0 C9                ret
0  0BC1 64756D6D79  db  'dummy filler   '
0       2066696C6C
0       6572202020
0  0BD0 0D         st2: dcr  c
0  0BD1 C26409             jnz  stst
0  0BD4 C36D09             jmp  ret1
1               ;
```

```
1                ;
1                dmfl: equ 4017h
1                dfl2: equ 4018h
1                excf: equ 3ea0h
1                ;
1                ;
1                ;  last segment of v2.0,to log the
1                ;scans etc..
o1                org  0c00h
0   0C00 010107      lgsc:  lxi  b,0701h
0   0C03 CD4B08             call  rcl1
0   0C06 3A903D             lda   3d90h
0   0C09 FE00               cpi  0
0   0C0B C2310C             jnz  lgs1
0   0C0E 3A1040             lda  mnfl
0   0C11 FE00               cpi 0
0   0C13 C2310C             jnz lgs1
0   0C16 3E01               mvi  a,1
0   0C18 321040             sta  mnfl
0   0C1B CDAD0D             call  lgdm
0   0C1E 01060C       lgdt:  lxi    b,0c06h   log the date
0   0C21 CD4B08             call rcl1
0   0C24 3EB3               mvi  a,0b3h
0   0C26 CD700C             call  loga
0   0C29 CDF60C             call  gt2b
0   0C2C 0E02               mvi  c,02
0   0C2E CDAC0C             call  lgt
0   0C31 010607       lgs1:  lxi   b,0706h
0   0C34 CD4B08             call rcl1
0   0C37 3A903D             lda  3d90h
0   0C3A FE01               cpi  01
0   0C3C C2440C             jnz  lgs2
0   0C3F 3E00               mvi  a,0
0   0C41 321040             sta  mnfl
0   0C44 3EA3         lgs2:  mvi  a,0a3h
0   0C46 CD700C             call loga
0   0C49 CDD10C             call gt3b
0   0C4C 0E03               mvi  c,03
0   0C4E CDAC0C             call lgt
0   0C51 CD530D       lgnt:  call cksn   see if the sensors
0   0C54 3E04                mvi  a,04       have chanhged
0   0C56 F5           lgs3:  push  psw
0   0C57 CDC609              call gtht
0   0C5A DA600C              jc  lgs4
0   0C5D CDE80C              call lghl
0   0C60 F1           lgs4:  pop  psw
0   0C61 3C                  inr  a
0   0C62 FE08                cpi 08
0   0C64 C2560C              jnz  lgs3
0   0C67 3E0F                mvi  a,0fh
0   0C69 CD700C              call loga
0   0C6C CDB90C              call ceof
0   0C6F C9                  ret
1                ;
1                ;   Put the acc. in the store
1                ;
```

```
0  0C70 D5          loga:  push  d
0  0C71 E5                 push  h
0  0C72 F5                 push psw
0  0C73 211140             lxi   h,ptr1
0  0C76 56                 mov   d,m
0  0C77 23                 inx   h
0  0C78 5E                 mov   e,m
0  0C79 3E80               mvi   a,80h
0  0C7B BA                 cmp   d
0  0C7C DA9D0C             jc    badp   ;check for bad store add.
0  0C7F 3E40               mvi   a,40h
0  0C81 BA                 cmp   d
0  0C82 D29D0C             jnc   badp
0  0C85 3E00               mvi   a,0
0  0C87 BB                 cmp   e
0  0C88 C2940C             jnz   log1
0  0C8B 3E80               mvi   a,80h
0  0C8D BA                 cmp   d
0  0C8E C2940C             jnz   log1
0  0C91 110041             lxi   d,4100h
0  0C94 F1          log1:  pop   psw
0  0C95 12                 stax   d
0  0C96 13                 inx   d
0  0C97 73                 mov   m,e
0  0C98 2B                 dcx   h
0  0C99 72                 mov   m,d
0  0C9A E1                 pop   h
0  0C9B D1                 pop   d
0  0C9C C9                 ret
0  0C9D 3E41        badp:  mvi   a,41h
0  0C9F 57                 mov   d,a
0  0CA0 321140             sta   ptr1
0  0CA3 3E00               mvi   a,0
0  0CA5 5F                 mov   e,a
0  0CA6 321240             sta   ptr2
0  0CA9 C3940C             jmp   log1
0  0CAC 11803D      lgt:   lxi   d,3d80h
0  0CAF 1A          lgt1:  ldax   d
0  0CB0 CD700C             call  loga
0  0CB3 13                 inx   d
0  0CB4 0D                 dcr   c
0  0CB5 C2AF0C             jnz   lgt1
0  0CB8 C9                 ret
0  0CB9 3E0E        ceof:  mvi   a,0eh
0  0CBB CD700C             call  loga
0  0CBE CD700C             call  loga
0  0CC1 CD700C             call  loga
0  0CC4 211140             lxi   h,ptr1
0  0CC7 56                 mov   d,m
0  0CC8 23                 inx   h
0  0CC9 5E                 mov   e,m
0  0CCA 1B                 dcx   d
0  0CCB 1B                 dcx   d
0  0CCC 1B                 dcx   d
0  0CCD 73                 mov   m,e
0  0CCE 2B                 dcx   h
```

```
0  0CCF 72                    mov  m,d
0  0CD0 C9                    ret
0  0CD1 11803D       gt3b:    lxi  d,3d80h
0  0CD4 21903D                lxi  h,3d90h
0  0CD7 0E03                  mvi  c,3
0  0CD9 7E           gtb1:    mov  a,m
0  0CDA 07                    rlc
0  0CDB 07                    rlc
0  0CDC 07                    rlc
0  0CDD 07                    rlc
0  0CDE 23                    inx  h
0  0CDF 86                    add  m
0  0CE0 12                    stax  d
0  0CE1 13                    inx  d
0  0CE2 23                    inx  h
0  0CE3 0D                    dcr  c
0  0CE4 C2D90C                jnz  gtb1
0  0CE7 C9                    ret
0  0CE8 3EF2         lghl:    mvi  a,0f2h
0  0CEA CD700C                call loga
0  0CED 7C                    mov  a,h
0  0CEE CD700C                call loga
0  0CF1 7D                    mov  a,l
0  0CF2 CD700C                call loga
0  0CF5 C9                    ret
0  0CF6 11803D       gt2b:    lxi  d,3d80h
0  0CF9 21903D                lxi  h,3d90h
0  0CFC 0E02                  mvi  c,2
0  0CFE 7E           gtb2:    mov  a,m
0  0CFF 07                    rlc
0  0D00 07                    rlc
0  0D01 07                    rlc
0  0D02 07                    rlc
0  0D03 23                    inx  h
0  0D04 86                    add  m
0  0D05 12                    stax  d
0  0D06 13                    inx  d
0  0D07 23                    inx  h
0  0D08 23                    inx  h
0  0D09 0D                    dcr c
0  0D0A C2FE0C                jnz  gtb2
0  0D0D C9                    ret
0  0D0E C3A00E       cksc:    jmp  ckn
0  0D0F 12                    db  12h
0  0D12 3E20                  mvi  a,20h
0  0D14 D312                  out pc
0  0D16 DB10                  in pa
0  0D18 E602                  ani  2
0  0D1A C8                    rz
0  0D1B 3E22                  mvi  a,22h
0  0D1D D312                  out  pc
0  0D1F AF                    xra  a
0  0D20 D312                  out  pc
0  0D22 CD510C                call lgnt
0  0D25 C9                    ret
0  0D26 D5           snlg:    push  d
```

```
0  0D27 E5                    push  h
0  0D28 110404                lxi   d,0404h
0  0D2B 3EA2          snl1:   mvi   a,0a2h
0  0D2D CD700C                call  loga
0  0D30 7A                    mov   a,d
0  0D31 07                    rlc
0  0D32 07                    rlc
0  0D33 07                    rlc
0  0D34 07                    rlc
0  0D35 C602                  adi   02
0  0D37 D312                  out   pc
0  0D39 3E10                  mvi   a,10h
0  0D3B D311                  out   pb
0  0D3D DB10                  in  pa
0  0D3F CD700C                call  loga
0  0D42 3E11                  mvi   a,11h
0  0D44 D311                  out   pb
0  0D46 DB10                  in pa
0  0D48 CD700C                call  loga
0  0D4B 14                    inr   d
0  0D4C 1D                    dcr   e
0  0D4D C22B0D                jnz   snl1
0  0D50 E1                    pop   h
0  0D51 D1                    pop   d
0  0D52 C9                    ret
0  0D53 E5            cksn:   push  h
0  0D54 213040                lxi   h,4030h
0  0D57 3E42                  mvi   a,42h
0  0D59 CD680D                call  cks2
0  0D5C 3E52                  mvi   a,52h
0  0D5E DC680D                cc cks2
0  0D61 3E62                  mvi   a,62h
0  0D63 CD900E                call  smwer
0  0D66 E1                    pop   h
0  0D67 C9                    ret
0  0D68 D312          cks2:   out   pc
0  0D6A 3E10                  mvi   a,10h
0  0D6C D311                  out   pb
0  0D6E DB10                  in  pa
0  0D70 BE                    cmp   m
0  0D71 C2820D                jnz   cks3
0  0D74 23                    inx   h
0  0D75 3E11                  mvi   a,11h
0  0D77 D311                  out   pb
0  0D79 DB10                  in pa
0  0D7B BE                    cmp   m
0  0D7C C2820D                jnz   cks3
0  0D7F 23                    inx   h
0  0D80 37                    stc
0  0D81 C9                    ret
0  0D82 CD260D        cks3:   call  snlg
0  0D85 D5                    push  d
0  0D86 110404                lxi   d,0404h
0  0D89 213040                lxi   h,4030h
0  0D8C 7A            cks4:   mov   a,d
0  0D8D 07                    rlc
```

```
0  0D8E 07                    rlc
0  0D8F 07                    rlc
0  0D90 07                    rlc
0  0D91 C602                  adi 2
0  0D93 D312                  out  pc
0  0D95 3E10                  mvi  a,10h
0  0D97 D311                  out  pb
0  0D99 DB10                  in  pa
0  0D9B 77                    mov  m,a
0  0D9C 23                    inx  h
0  0D9D 3E11                  mvi  a,11h
0  0D9F D311                  out  pb
0  0DA1 DB10                  in  pa
0  0DA3 77                    mov  m,a
0  0DA4 23                    inx  h
0  0DA5 14                    inr  d
0  0DA6 1D                    dcr  e
0  0DA7 C28C0D                jnz  cks4
0  0DAA D1                    pop  d
0  0DAB B7                    ora  a
0  0DAC C9                    ret
0  0DAD AF              lgdm: xra  a
0  0DAE CD660E                call stfl
0  0DB1 321840                sta  dfl2
0  0DB4 322340                sta  dfl3
0  0DB7 322440                sta  vsfl
0  0DBA 3ED3                  mvi  a,0d3h
0  0DBC CD700C                call loga
0  0DBF 0E09                  mvi  c,9
0  0DC1 211A40                lxi  h,401ah
0  0DC4 7E              lgd1: mov  a,m
0  0DC5 CD700C                call loga
0  0DC8 23                    inx  h
0  0DC9 0D                    dcr  c
0  0DCA C2C40D                jnz  lgd1
0  0DCD C9                    ret
0  0DCE 3A2340          dpr: lda  dfl3
0  0DD1 FE01                  cpi 01
0  0DD3 C8                    rz
0  0DD4 010607                lxi  b,0706h
0  0DD7 CD4B08                call rcl1
0  0DDA 112040                lxi  d,4020h
0  0DDD CD240E                call svck
0  0DE0 3E01                  mvi  a,1
0  0DE2 322340                sta  dfl3
0  0DE5 C9                    ret
0  0DE6 CD6D0E          dpir: call dprn
0  0DE9 FE01                  cpi 1
0  0DEB C9                    ret
0  0DEC 3A1840                lda dfl2
0  0DEF FE01                  cpi 01
0  0DF1 CA050E                jz  dpi2
0  0DF4 010607                lxi  b,0706h
0  0DF7 CD4B08                call rcl1
0  0DFA 111A40                lxi  d,401ah
0  0DFD CD240E                call svck
```

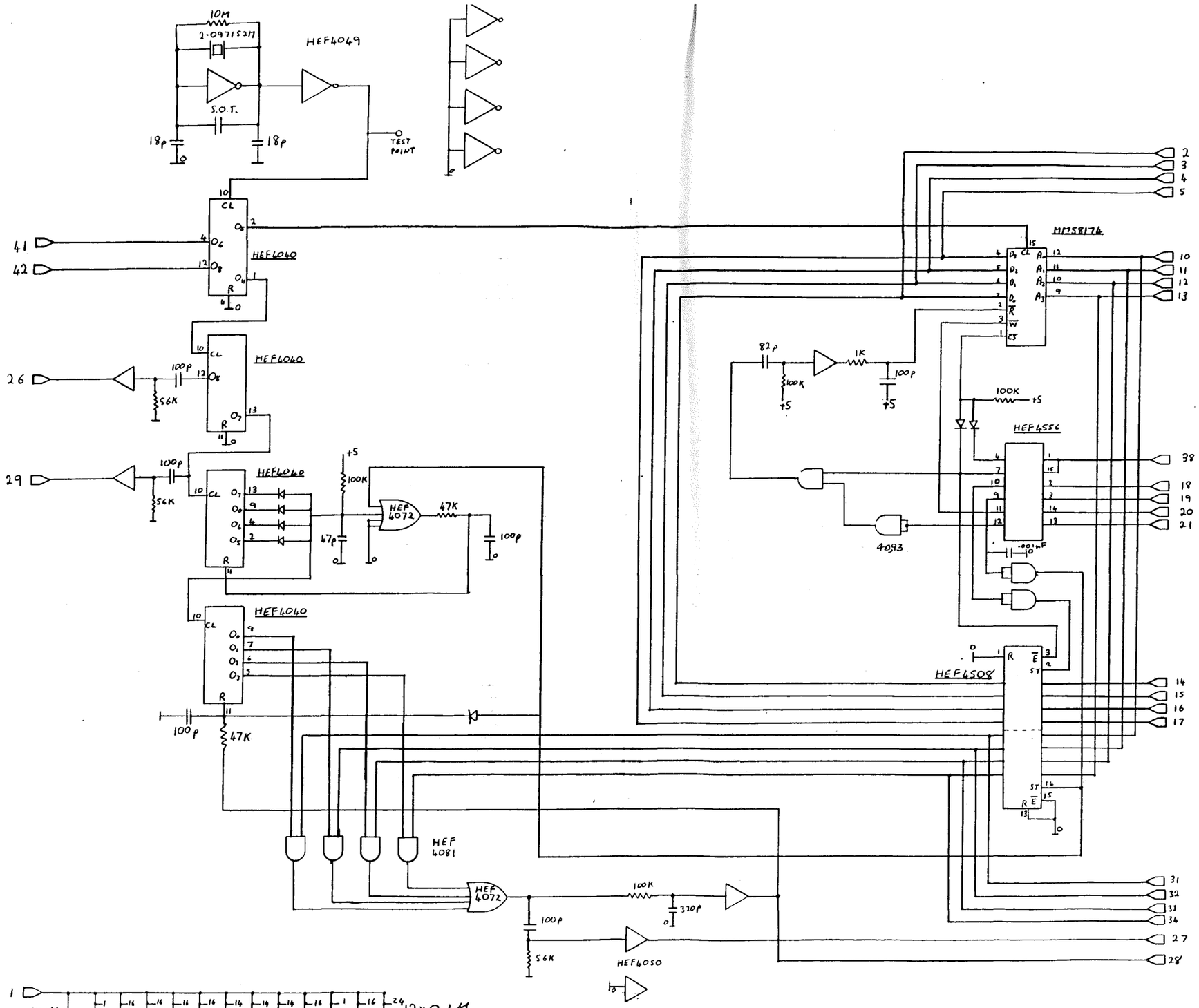
```
0  0E00 3E01              mvi  a,1
0  0E02 321840            sta  dfl2
0  0E05 010607     dpi2:   lxi  b,0706h
0  0E08 CD4B08            call rcl1
0  0E0B 111D40            lxi  d,401dh
0  0E0E CD240E            call svck
0  0E11 211D40            lxi  h,401dh
0  0E14 3A1A40            lda  401ah
0  0E17 C601              adi 1
0  0E19 BE                cmp  m
0  0E1A FA1E0E            jm   dpi3
0  0E1D C9                ret
0  0E1E 3E01       dpi3:  mvi  a,1
0  0E20 321740            sta  dmfl
0  0E23 C9                ret
0  0E24 21903D     svck:  lxi  h,3d90h
0  0E27 0E03              mvi  c,03
0  0E29 7E         svc1:  mov  a,m
0  0E2A 07                rlc
0  0E2B 07                rlc
0  0E2C 07                rlc
0  0E2D 07                rlc
0  0E2E E6F0              ani  0f0h
0  0E30 23                inx  h
0  0E31 86                add  m
0  0E32 23                inx  h
0  0E33 12                stax  d
0  0E34 13                inx  d
0  0E35 0D                dcr  c
0  0E36 C2290E            jnz  svc1
0  0E39 C9                ret
0  0E3A 3EC3       lgcs:  mvi  a,0c3h
0  0E3C CD700C            call loga
0  0E3F 110607            lxi  d,0706h
0  0E42 CD4B08            call rcl1
0  0E45 CDD10C            call gt3b
0  0E48 0E03              mvi  c,3
0  0E4A CDAC0C            call lgt
0  0E4D C9                ret
0  0E4E 3A2440     lgvs:  lda  vsfl
0  0E51 FE01              cpi  1
0  0E53 C8                rz
0  0E54 3E01              mvi  a,1
0  0E56 322440            sta  vsfl
0  0E59 010607            lxi  b,0706h
0  0E5C CD4B08            call rcl1
0  0E5F 111D40            lxi  d,401dh
0  0E62 CD240F            call svck
0  0E65 C9                ret
0  0E66 3E0A       stfl:  mvi  a,0ah
0  0E68 321740            sta  dmfl
0  0E6B AF                xra  a
0  0E6C C9                ret
0  0E6D 3A1740     dprn:  lda  dmfl
0  0E70 3D                dcr  a
0  0E71 C8                rz
```

```
0  0E72 321740              sta  dmfl
0  0E75 3EC3                mvi  a,0c3h
0  0E77 CD700C              call loga
0  0E7A 010607              lxi  b,0706h
0  0E7D CD4B08              call rcl1
0  0E80 CDD10C              call gt3b
0  0E83 0E03                mvi  c,3
0  0E85 CDAC0C              call lgt
0  0E88 CDB90C              call ceof
0  0E8B C9                  ret
0  0E8C 20202020     db '    '
0  0E90 DC680D      smwer:  cc  cks2
0  0E93 3E72                mvi  a,72h
0  0E95 DC680D              cc  cks2
0  0E98 C9                  ret
0  0E99 3132333435   db  '1234567'
0       3637
0  0EA0 F5          ckn:  push  psw
0  0EA1 E5                 push h
0  0EA2 D5                 push  d
0  0EA3 C5                 push  b
0  0EA4 3E28               mvi  a,28h
0  0EA6 D312               out  pc
0  0EA8 3E20               mvi  a,20h
0  0EAA D312               out  pc
0  0EAC DB10               in  pa
0  0EAE E602               ani  02
0  0EB0 CABD0E             jz  ckn2
0  0EB3 3E22               mvi  a,22h
0  0EB5 D312               out  pc
0  0EB7 AF                 xra  a
0  0EB8 D312               out  pc
0  0EBA CD510C             call  lgnt
0  0EBD C1          ckn2:  pop  b
0  0EBE D1                 pop  d
0  0EBF E1                 pop  h
0  0EC0 F1                 pop psw
0  0EC1 C9                 ret
1                  ;
1                  ;
1                  ;
1                  pc:  equ 12h
1                  dmfl:  equ  4017h
1                  dfl2:  equ  4018h
1                  dfl3:  equ  4023h
1                  ptr1:  equ  4011h
1                  ptr2:  equ  4012h
1                  mnfl:  equ  4010h
1                  vsfl:  equ  4024h
2                        Symbol Table
A21       06C7 A2        06B6 A3        06E9 ADD       0A7E APRT      072F
B3        0679 BADP      0C9D BAK1      01EC BAKUP     01E8 BDCH      0B84
BIL1      014B BIL2      014E BIL3      0176 BILN1     08C2 BILN2     08C5
BILN3     08ED BILN      08BA BIL       0143 BLH1      0B54 BNFL      4015
C0        0917 C3        069E CEOF      0CB9 CHEK      0A74 CHOK      0B7C
CI2       07C3 CI3       0777 CIN       0BB5 CI        001B CKN2      0EBD
```

CKN	0EA0	CKS2	0D68	CKS3	0D82	CKS4	0D8C	CKSC	0D0E
CKSN	0D53	CMST	084E	COM1	0839	COM2	083A	COM4	0840
COMD	0811	COMS	0814	COMT	0829	CON1	0585	CON2	0582
COS	052C	CO	019C	CRL1	0673	CRLF2	090C	CRLF	0191
CR	000D	CTRL	0013	D3	0783	DCXP	076E	DFL2	4018
DFL3	4023	DL1	013E	DLY1	013C	DMFL	4017	DPI2	0E05
DPI3	0E1E	DPIR	0DE6	DPRN	0E6D	DPR	0DCE	DXP1	07B0
DXPT	07AE	ERC1	063D	ERCH	063B	ERGT	0949	EROK	0664
EROR	0186	ERROR	0901	ESC	001B	EXCF	3EA0	F2	06EF
FMD1	05C9	FMD2	05EB	FMDN	05C6	GT2B	0CF6	GT3B	0CD1
GTB1	0CD9	GTB2	0CFE	GTH1	09DD	GTH2	09E4	GTHT	09C6
GTP1	092A	GTP2	0930	GTPM	0922	H1	01AD	H4	01BC
HIGHT	01A7	INC	0AFA	INIT	0B1C	INXP	0765	IXP1	07A4
IXPT	07A2	LCS1	09A4	LCS2	09BD	LCST	09A1	LGCS	0E3A
LGD1	0DC4	LGDM	0DAD	LGDT	0C1E	LGHL	0CE8	LGNT	0C51
LGS1	0C31	LGS2	0C44	LGS3	0C56	LGS4	0C60	LGSC	0C00
LGT1	0CAF	LGT	0CAC	LGVS	0E4E	LOG1	0C94	LOGA	0C70
LPT1	0B8C	LPTR	0B4C	LTR	07BA	MEMT	0080	MJER	066A
MNFL	4010	MNUS	0AC3	MULT	0A5C	NE	0AED	NOBAK	0209
PA	0010	PBIN	0532	PBN1	0534	PBN2	0543	PB	0011
PC	0012	PMPT	0800	PMT	008F	PRAC	0741	PRAD	04F2
PRD1	0501	PRD2	050D	PRD3	050F	PRD4	051A	PRD5	0520
PRI	0B04	PRT	0ACC	PSHT	020D	PSV1	4013	PSV2	4014
PSV3	400E	PSV4	400F	PSW1	00D6	PSW2	00E9	PSW3	00F7
PSW4	00FE	PSW5	0113	PSWD	00D1	PSWM	0132	PSXC	0136
PTR1	4011	PTR2	4012	PUTD	075E	PUTH	0752	PWDN	0016
PWOK	012F	PWR1	0122	PWUP	0000	PWWR	010D	QUAD	0A08
RAPT	073E	RCFT	089D	RCL1	084B	RCLK	099A	RCL	099D
RD	08AC	RET1	096D	RET	087A	RHLL	0B97	RIS1	0041
RIS2	0092	RIS3	00A1	RIS4	00B0	RIS5	00BF	RIS6	00CE
RIS	0027	RT	088D	SAD1	0A91	SAD2	0AA4	SAPT	0730
SCARD	0003	SCLK	094F	SD	0884	SFLG	3F08	SIS	4025
SI	0B08	SMWER	0E90	SNL1	0D2B	SNLG	0D26	SPCE	0729
SPRT	0AB5	SPTR	0B3D	SP	0020	ST2	0BD0	STACKT	3CFA
STFL	0E66	STRBOT	0041	STRTOP	0060	STRT	0A5F	STST	0964
ST	087B	SUB	0AEB	SVC1	0E29	SVCK	0E24	TAS1	05F9
TAS2	0614	TASC	05FC	TB2	0593	TBIN	054B	TBN1	055A
TBN2	0589	TBN3	05B5	TBN4	05C0	TPRT	0703	TR1	0405
TR21	045D	TR51	0490	TR52	04A1	TRL1	0439	TRL2	0447
TRL3	0463	TRL4	0471	TRL5	0480	TR	0400	UC	00ED
UD	00EC	VSFL	4024	WAIT	0972	WCDB	0981	WCLK	0987
XCOK	012A								

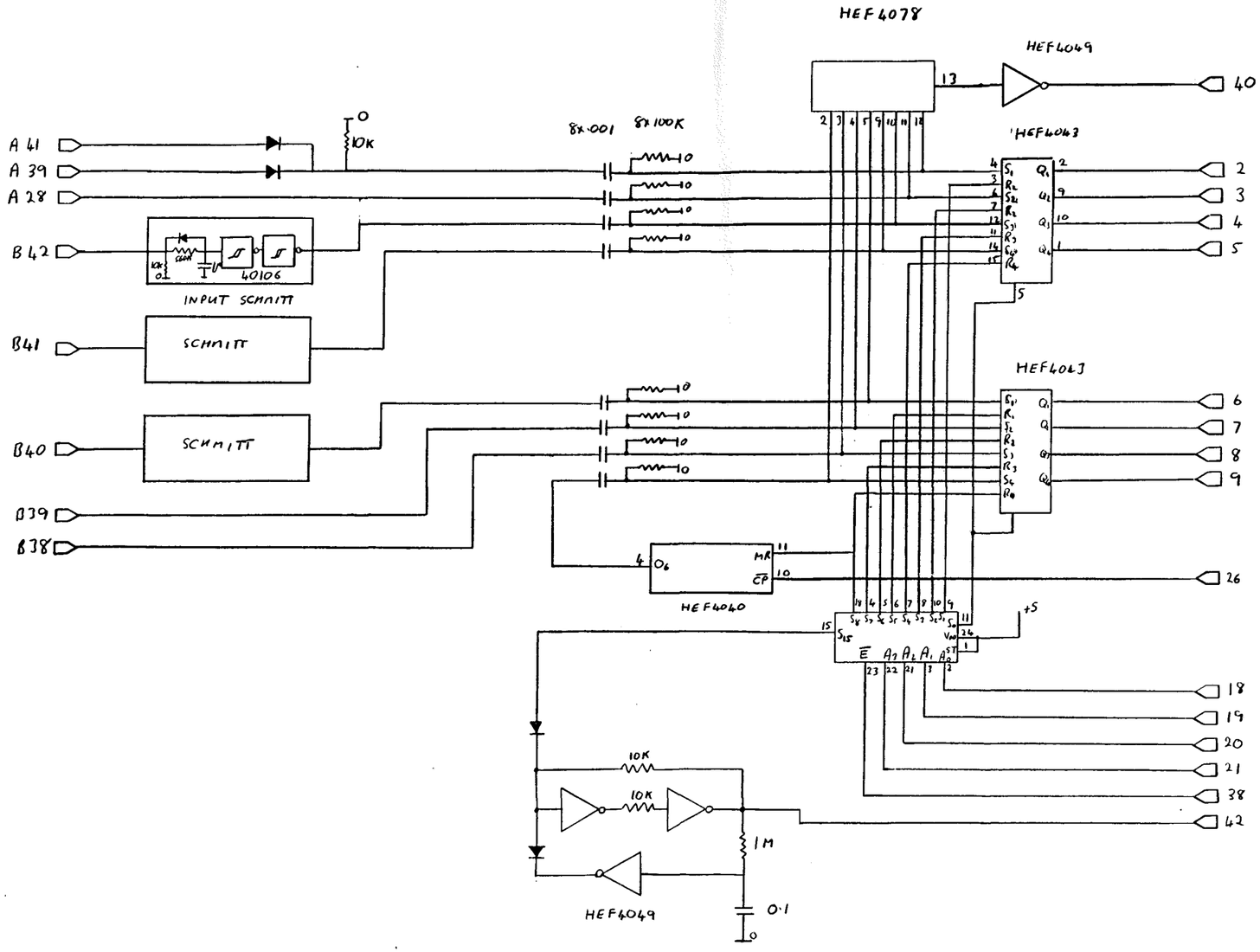
APPENDIX 4

Circuit diagrams.

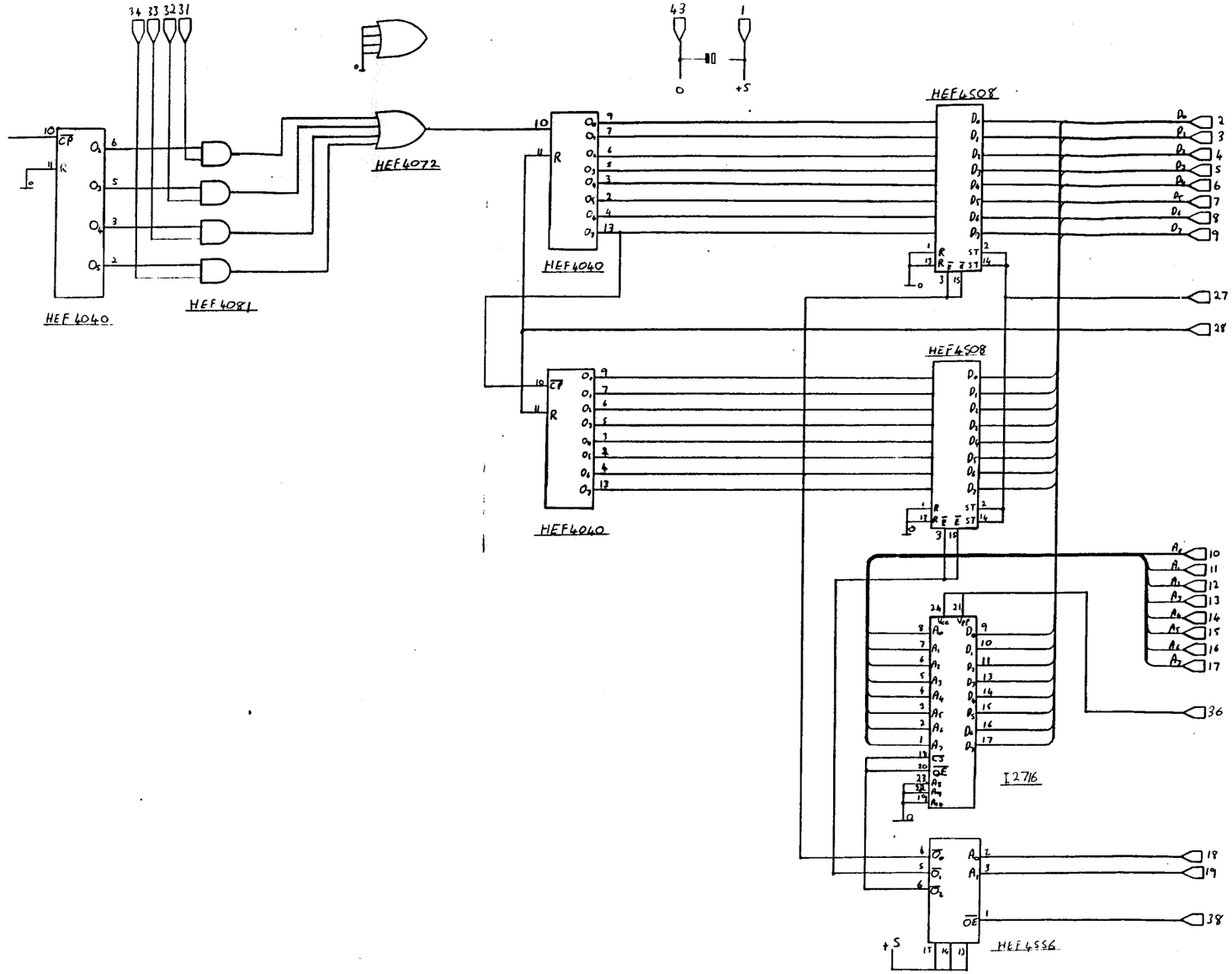


DATARING CLOCK/SAMPLE PULSE BOARD

R PALIN
3 AUG 1982

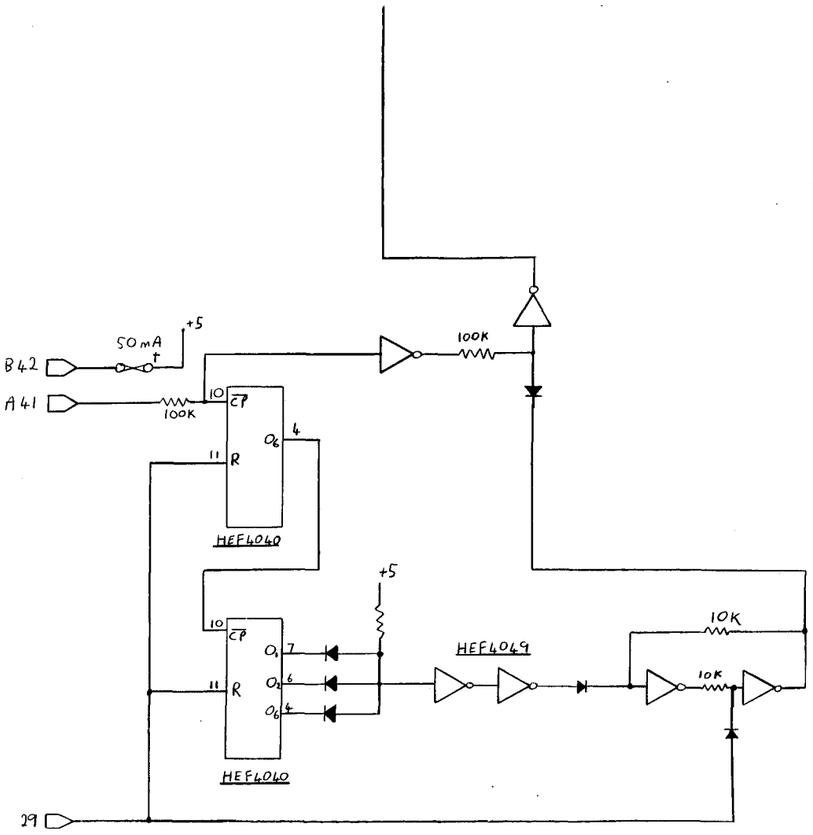


DATARING START-UP BOARD
R PALIN OCT 1982



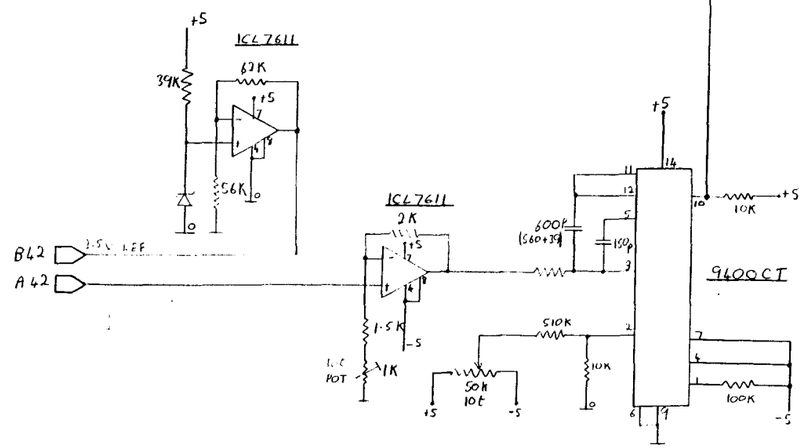
DATARING SENSOR BOARD

R. PALIN 5th AUG 1982

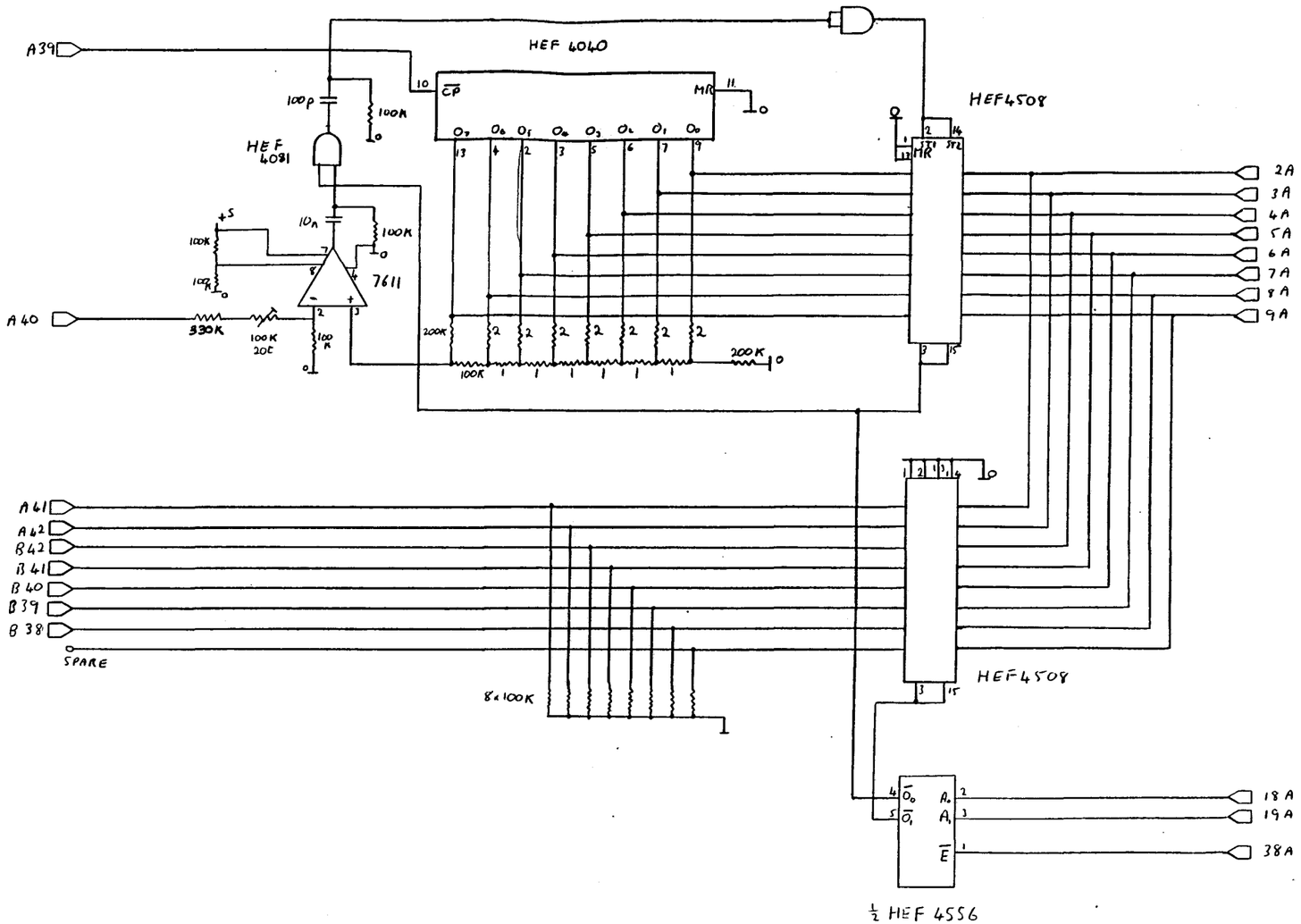


† 20mm FUSE IN RS 412-784 CLIPS

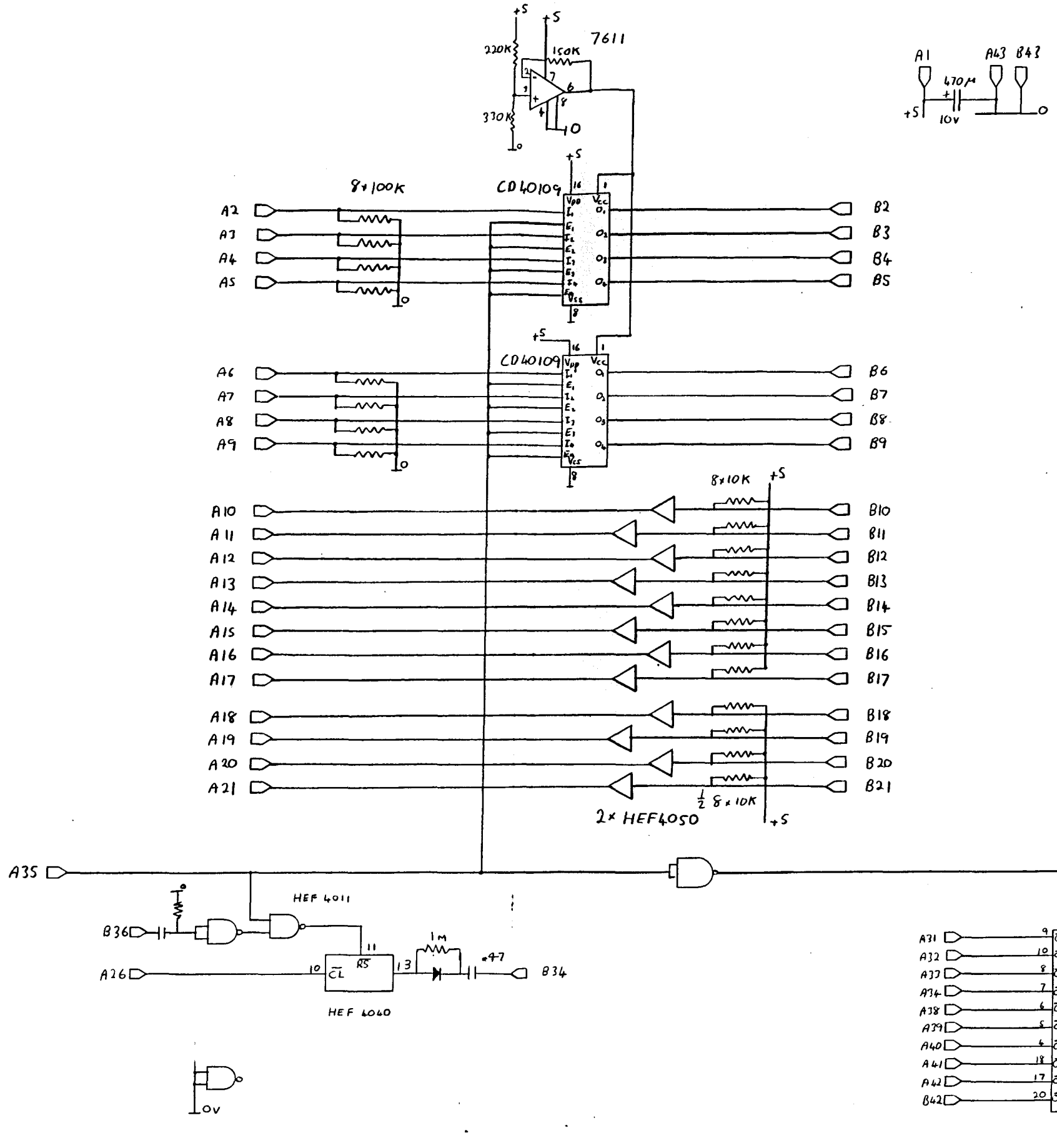
DIGIQUART2 INTERFACE



OTT INERFACE



DATARING D.V.M./STATUS BOARD
DRAWN RIR PALIN
OCT 1982

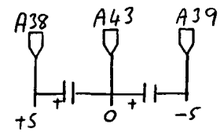


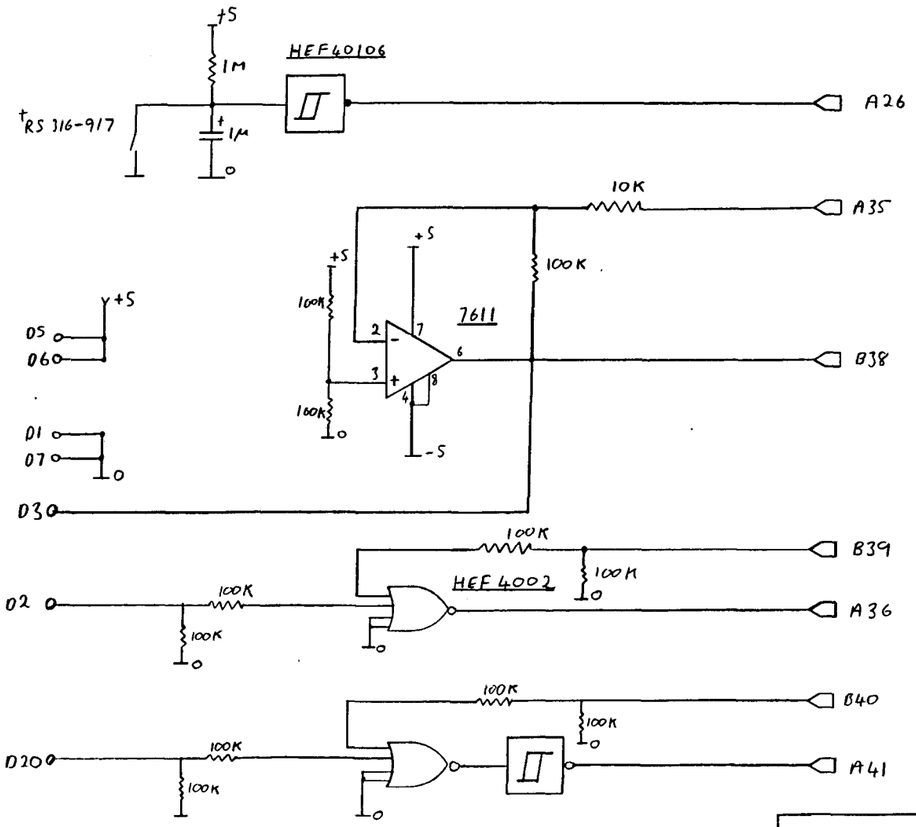
DATARING BUFFER BOARD
DRAWN BY R. PALIN
24th SEPT 1982

8085 CPU PCB
Circuit Schematic

INSTITUTE OF OCEANOGRAPHIC SCIENCES BIDSTON OBSERVATORY

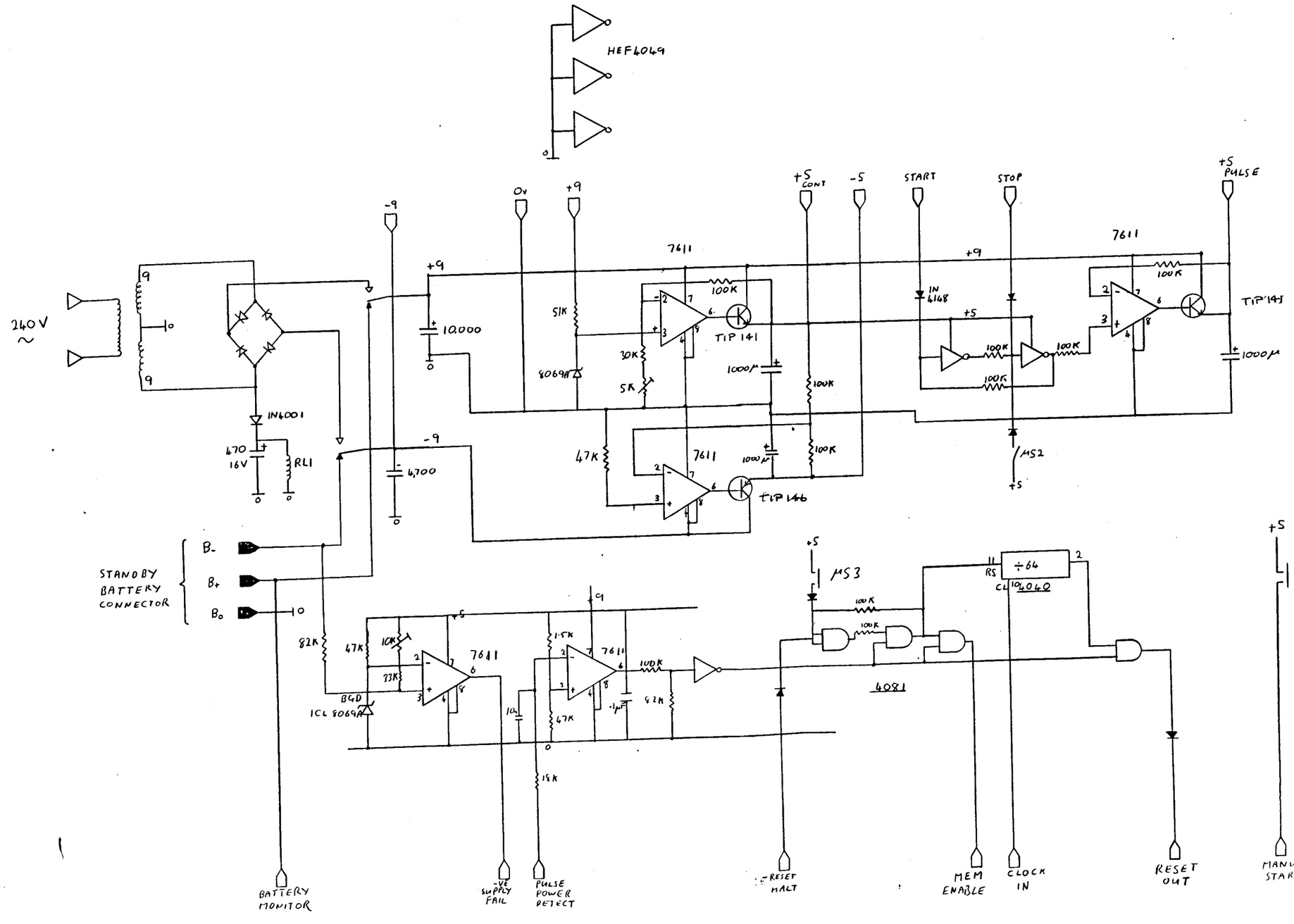
DRAWN PHOTONIC	ALL DIMNS IN	ANGLE PROJECTION
DATE 7 JUNE 82	TITLE	
APPROVED		
SCALE		
MATERIAL	SHEET OF SHEETS	DRG. NO.





† NOTE: MOUNT AT FRONT OF CARD

DATARING RS232 INTERFACE
DRAWN R PALIN OCT. 1982



DATARING POWER SUPPLY BOARD

RIR PALIN SEPT 1982