Nuclear Instruments and Methods 187 (1981) 435–441
North-Holland Publishing Company

AN AUTONOMOUS CAMAC CRATE CONTROLLED BY A PERSONAL COMPUTER

P. KLOPF, L.A. VAN RHEE and W. STÜBER
Commission of the European Communities, Joint Research Centre, Central Bureau for Nuclear Measurements, B.2440, Geel, Belgium

Received 4 December 1980

A low cost version of an autonomous CAMAC crate has been realised using the personal microcomputer TRS 80 for programmed control. To keep the hardware development and production effort low, the serial CAMAC crate controller type L2 is used. In this case the necessary interface may be a very simple and inexpensive one. The CAMAC crate is equipped for a scaler experiment only, but extensions for the addition of a multichannel analyser in the same experiment are already available. The program is written in BASIC extended by a general CAMAC statement and some other useful expansions to the TRS-DISK BASIC.

1. Introduction

CAMAC is a well known and widely accepted standard for digital interfacing of physics experiments to computers. Nowadays it is also used in other fields, such as industrial control and medical electronics. As CAMAC is only an interface, a CAMAC crate or a group of crates must be controlled by a digital computer, which eventually may be situated in the crate controller itself.

As an outgrowth of microelectronics computer prices have diminished drastically during the last decade. The last step in this evolution is at present the personal microcomputer, which due to mass-production and stiff competition is so inexpensive that it would be more economic to control a CAMAC crate by such a computer than by a central computer or by one of those special computerized crate controllers. Most personal computers have relatively good operating systems and other system and high level language software, often originated by independent software houses.

Although there exist several computer languages specifically designed for control and data acquisition purposes, the general language BASIC is preferred in small systems with simple applications, because it is easy to learn. So the users themselves, who are specialists in their field but not in computer programming, can extend existing user programs and even write new ones. For this reason BASIC is used in our application, too.

One of the most widespread and inexpensive personal computers with ample software resources and a good BASIC is the Radio Shack TRS-80. So we decided to use this microcomputer to control a nuclear physics experiment with up to 27 6-decade CAMAC scalers and one timer. Our aim was not only to get economy in price, but also in hardware development and production work. By the latter goal direct interfacing of the microcomputer to the crate dataway was excluded, and a standard CAMAC crate controller had to be used.

There are two distincly different types of crate controllers, the type A1 or A2 controller to be connected to the all-parallel Branch Highway, and the type L2 controller to be connected to the byte-organised Serial Highway. It turned out that the serial crate controller (SCC) only needs a very simple interface for a connection to the microcomputer, despite the fact that the signal levels are non-TTL. So we decided to use the standard L2 controller, notwithstanding its higher price.

It should be emphasised, that according to the similarity of the input–output system of most microcomputers, our TRS-80 to CAMAC interface could be used on most other types of microcomputers with minor or no changes at all.

2. The TRS-80 microcomputer

The minimum configuration of this personal computer, namely the Z80 microprocessor combined with

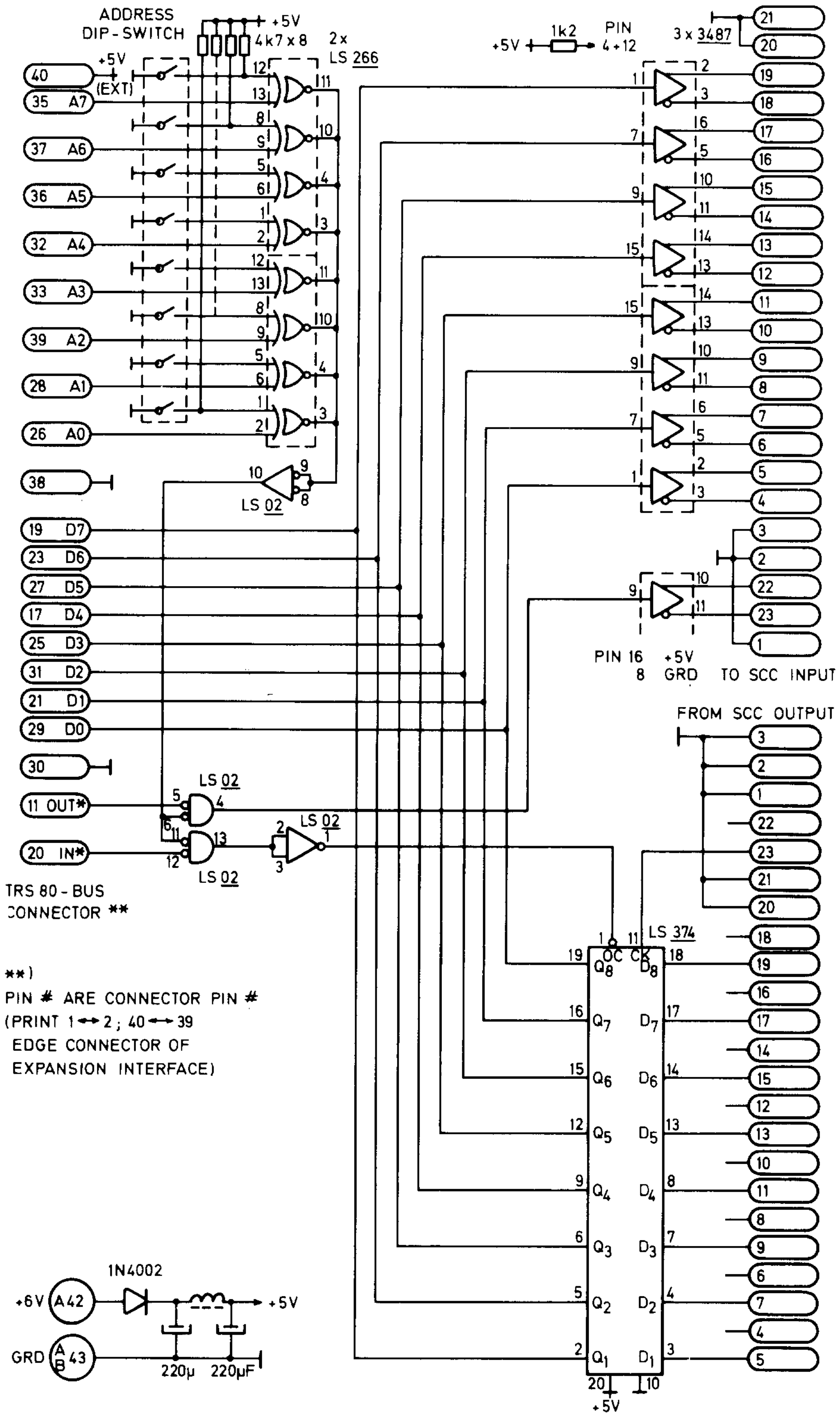

Fig. 1. TRS-80/CAMAC interface.

4 kbyte RAM and 4 kb or 12 kb BASIC in ROM in the keyboard unit, plus a television display and an audio cassette player, is completely insufficient for professional applications. Fortunately the TRS-80 can be upgraded by additional devices, mainly the expansion interface unit with extra memory and a floppy disk controller for connection of up to four (mini) floppy disk drives to the computer [1]. Replacing the cassette by floppy disk does not only centuple the speed of external memory, but its powerful disk operating system (DOS) eases program development and gives access to other high level languages as FORTRAN, PASCAL and COBOL. As with all personal computers the necessary programs can be developed and tested on the computer itself without the need for a costly development system.

External interfaces can be connected to the processor via the data-and-address-bus connector, but a disadvantage of the TRS-80 is that it can house and feed only one interface card. In our case this is not a serious drawback, because the interfaces to the 'real world' are housed in the CAMAC crate.

3. The hardware interface TRS-80/CAMAC

This unit acts as a normal TRS-80 port, which is addressed by the lower 8 address lines and an 'out' or an 'in' strobe pulse (fig. 1). If during an output operation the right address bit combination is on the address lines, the out strobe pulse from the microprocessor is gated onto the clock line to the serial crate controller. Thus the controller is stimulated to accept the data on the 8 data lines. In addition the controller places its own data onto its output lines and generates a clock pulse, too, which is used to strobe the controller data into an 8-bit register (74 LS 374) on the interface. The contents of this tri-state register can be read by a following input operation into the microprocessor. Thus an input operation must always be preceded by an output operation. This is mandatory by the CAMAC rules. Also mandatory is that data and clock to and from the serial crate controller are conveyed on 100 Ω lines according to the RS-422 standard. Nevertheless, for the short distances normally needed (<3 m), TTL-signals

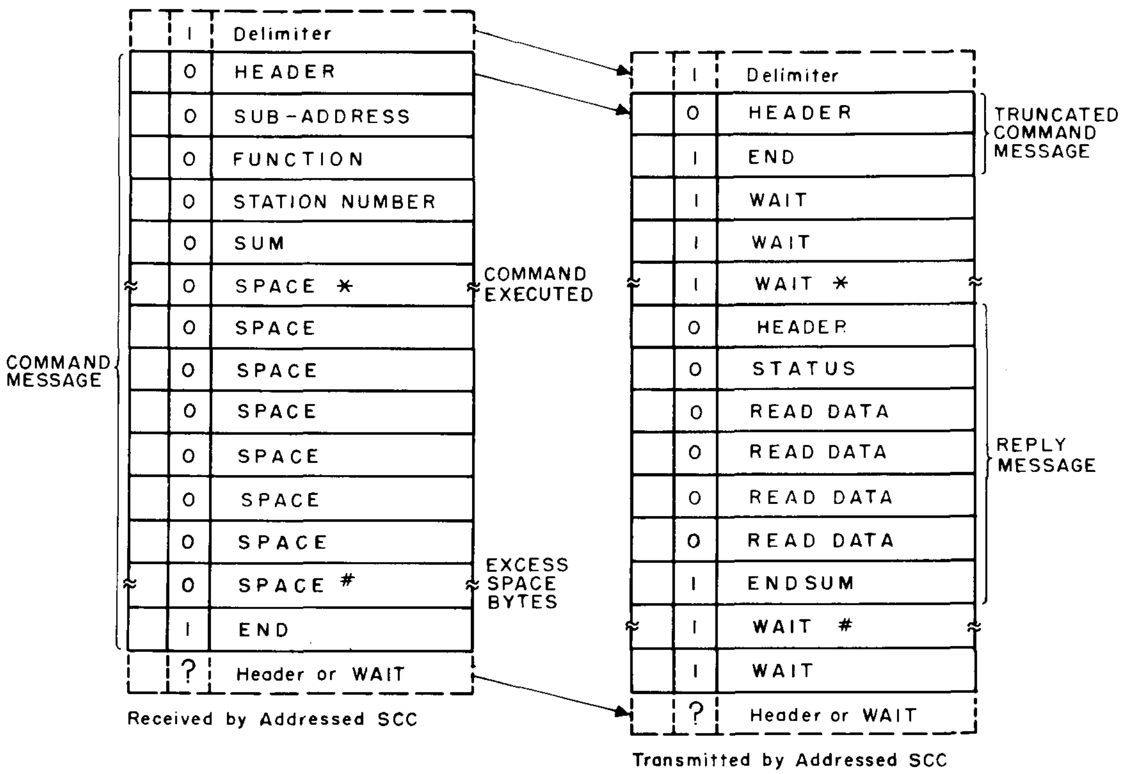

Fig. 2. Command/reply sequence: read operation, byte-serial mode.

could be used. This is exploited here because the crate controller used (Bi Ra Systems Inc.) generates its symmetrical output signals by TTL-drivers in push-pull. If we completely stick to CAMAC standards and use not only RS-422 line-drivers (e.g. Motorola MC3487) but also line receivers (e.g. Motorola MC3486) the distance between computer and CAMAC crate may be 1 km or more. Of course there may also be more than one crate connected to the serial highway (maximum 62), but in most cases this would mean a workload too heavy for the personal computer.

4. The software interface

The CAMAC serial highway is mainly intended for use on long interconnections exposed to electrical noise. Therefore error detection facilities have been provided, mainly byte-parity and column-parity checks. In addition up to 62 crates must be addressed separately; they must be supplied with commands and data, must deliver data and must have the possibility to generate program interrupts. All this requires a relatively complicated transmission protocol, which has been specified in a lengthy document [2], and which, in our case, must be observed by the CAMAC driver routine in the microcomputer program. This routine has been written in assembly language because in BASIC it would be inadmissably slow. To give an idea of the requirements to be fulfilled a data read operation sequence will be illustrated in short. It is shown schematically in fig. 2, which is reproduced from the above mentioned specification document.

Information is transferred serially byte after byte, but only 6 of the 8 bits of each byte carry the information proper. The leftmost bit of each byte cares for odd parity of the whole byte, and the second left bit acts as a delimiter bit to mark the beginning and end of a message. The "header" is simply a 6-bit crate address, which in our case is always "1". Then follow the sub-address, the function code and the station number of the addressed module in the crate. "Sum" and "endsum" is used for error detection and must conserve even column-parity over all bytes of the message from header byte to sum byte. Space bytes are "dummy" bytes which serve within a message for continuation of the data and clock stream; wait bytes do the same outside a message. The status byte reflects the state of the SCC status register. All these are necessary to transfer the 4×6 bits of the desired read data.

It is evident that the computer has to do a lot of bit handling, arranging 8-bit bytes into 6-bit ones and vice-versa, inserting and removing parity and delimiter bits and generating space-, end- and sumbytes. This is time-consuming, but not difficult. Even the generation of the parity bits is not difficult, because the Z80 processor generates a parity bit automatically during the execution of certain instructions, and because the column-parity byte can be generated by an "exclusive or" of all message bytes, which is in the instruction repertoire of the Z80.

The BASIC system program of the TRS-80 has been extended inter-alia by a general CAMAC call of the form

$$E = \mathrm{FN} * \mathrm{C}(N, A, F, V)$$

The letters between parentheses represent normal BASIC integer values, integer variables or expressions. As usual we have N = station number, A = subaddress, F = function code. V is the data element to be transferred, which may be an integer or floating point value or variable (single or double precision), or also a string. If V is preceded by a semicolon instead of a comma, it may be even an array, for fast transfer of a group of values. In E a numerical value between 0 and 6 is returned, which reflects the contents of the SCC status register. The details of all this and more may be found in a special paper [3].

Anyway, when during an execution of a BASIC program the CAMAC call is encountered, program control is transferred to the CAMAC driver routine, which interprets the NAF-code, executes the corresponding CAMAC command, and then returns to BASIC, where the returned variables may be treated further. According to the error detection possibilities of the serial highway extensive error checking is done. This might eventually be omitted to increase speed.

5. The realised CAMAC system

To get practical experience with a working and really useful but not too complicated system it was decided to automate first a nuclear physics measuring installation using only CAMAC scalers in connection with radiation detectors. Due to the modularity of CAMAC and to the general applicability of our design an extension to more demanding applications would not be difficult.

The TRS-80 system used here has one mini-floppy disk drive and 48 kb of random access memory. This memory size is too large for our application, but allows for comfortable programming and future extensions.

A block diagram of the whole CAMAC system is shown in fig. 3. In a minicrate with 17 module places up to 7 CAMAC 6-decade quad scalers will be supported by the BASIC user program. In connection with a quartz controlled clock generator the first of the up to 28 scalers is used as a programmable timer, which is preset by the computer according to the desired duration of a measurement run. The timer then starts and stops the run automatically via an inhibit line to all scaler inhibit inputs. During the measurement the momentary contents of all scalers are displayed on the display monitor of the TRS-80. At the end of each measurement run the computer stores all scaler data on the mini-floppy disk and eventually initiates another run. At the end of all runs the stored results can be printed out. Normally one would do this via the printer output of the TRS-80, but the users asked to use their existing teletypewriter, which had to be interfaced via an industrial CAMAC module.

Exploiting the fact that our personal computer system is dedicated to one well-defined measuring task, it was possible to make the operating job of the experimentalist very easy. When the mains power is switched on a bootstrap loader program in the read-only memory of the computer is started automatically and loads the disk operating system with CAMAC extensions from the floppy disk into the computer. The DOS then loads and starts the user BASIC program which after some initialization and test procedures offers the user on the display screen a "menu card" of possible actions (see flow diagram, fig. 4). Having keyed in, for example, the ciphers 00 (=prepare a new experiment) the user gets a series of questions on the display and must key in the needed parameters, such as identification code, number of scalers, duration of a run, number of runs. The following is an example of an experiment start procedure:

```
ENTER EXPERIMENT NUMBER (0000–9999)? 0001
NUMBER OF USED SCALER STATIONS (< = 7)? 3

!!! CLOCK = 100 HZ !!!
ENTER THE RUNTIME (SEC) (1 – 167772)? 1000
RUNS/EXPERIMENT (1–10)                ? 10

ENTERED EXP. PARAMETER VALID? ENTER Y OR N?
```

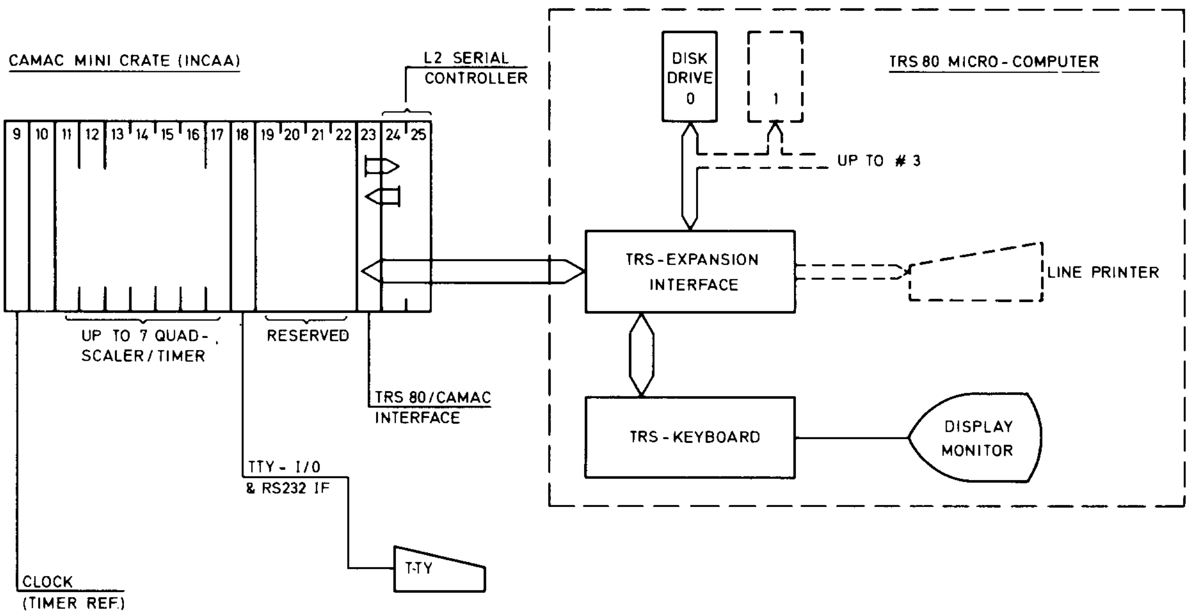

Fig. 3. TRS-80/CAMAC system.

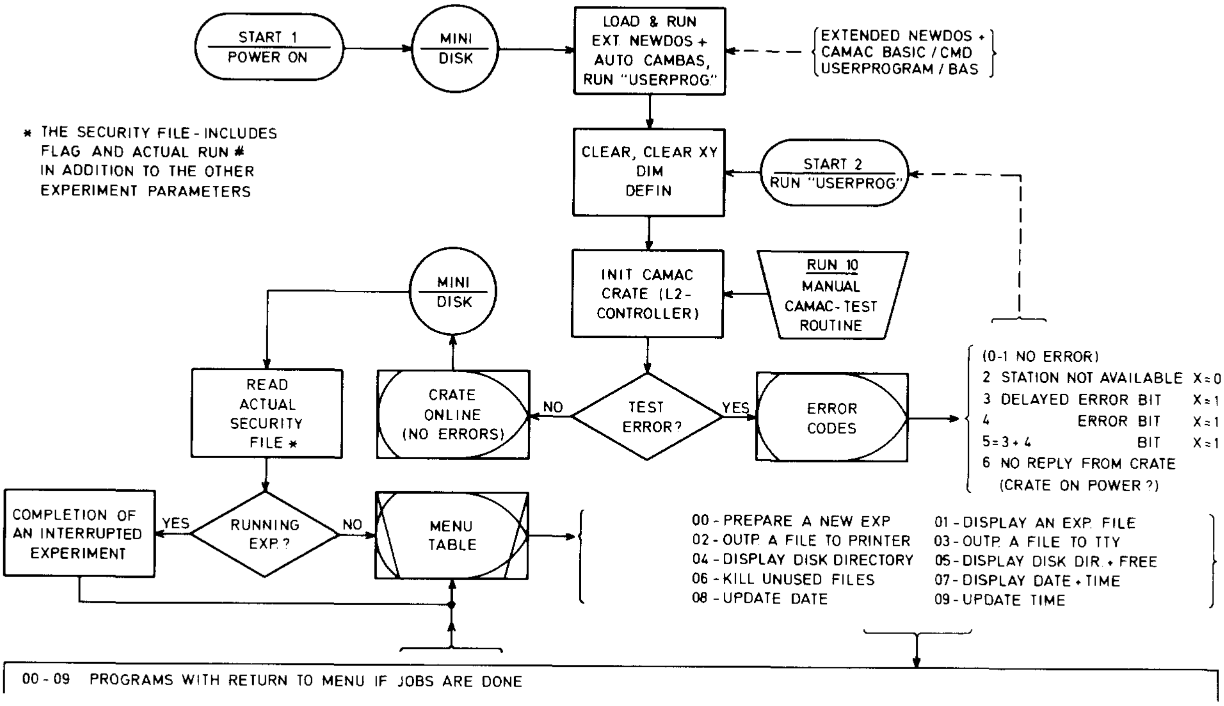

Fig. 4. TRS-80/CAMAC simplified flow diagram for start.

Having collected all parameters, the computer starts the measurement procedure and continues it for minutes, hours or days. During this time the keyboard is made ineffective by software to avoid disturbances by inadvertent keystrokes. The experiment parameters are not only recognised by the computer to set up the proper experiment conditions, but are also stored on disk in a separate security file, including an "experiment-is-running" flag. After each run the actual run number in this file is updated. Thus after a mains power interruption the self-restarting computer need only look up this file, and then can continue the experiment where it left off.

6. Experience and prospects

Since its delivery early in March 1980 the TRS-80/CAMAC system has worked almost continuously and quite satisfactorily in the Radionuclides Laboratory. During this time two hardware failures have occurred in CAMAC modules, none in the personal computer. One system collapse has occurred, because the DOS on the diskette was partially destroyed, possibly by a power failure during a disk access. The already low probability for such a fault would be further diminished considerably by using a second disk drive for the measurements results, whereas the system and user programs are on a write-protected diskette on the first drive. In our case it was possible to recover the accumulated measurement results by use of a new system diskette and a second disk drive.

For future applications the possibility must be provided to connect a multichannel analyser to the system. In the meantime this problem has already been solved for a non-CAMAC application by a direct connection to the RS-232 interface of the TRS-80 [4]. Only if there were sound reasons for a connection via a CAMAC module additional software development would be necessary.

The speed of CAMAC operations is not very impressive. It takes about 10 ms to execute one CAMAC read command transferring one 24-bit dataword. This is mainly due to the slow interpretation process in BASIC. If a group of data is transferred into an array the time per dataword is only 2.1 ms. The larger part of this time, viz. 1.3 ms, is needed for the transfer protocol (fig. 2), and it is estimated that the use of an all-parallel crate controller would reduce the time per dataword to about 1 ms. This time is not an absolute limit, because it is used up by software manipulations of the data, such as conversion from

24-bit integer to floating point. If it were for sheer speed, a buffer region of 256 bytes could be output to the CAMAC interface within 3 ms, i.e. 1 byte per 12 μs. Such a high speed would be needed, for example, for periodic refresh of a display. Of course, all these time values would be reduced further by use of a faster computer. The TRS-80 has a clock frequency of 1.77 MHz, whereas a Z80A processor with appropriate memory can run at 4 MHz. Anyway, for many practical applications the TRS-80 with its BASIC will be sufficiently fast.

When higher data rates occur or fast control by the microcomputer is indispensable, a BASIC compiler may be used to get a machine program running 3–10 times faster than the original BASIC. In this case it would also be advisable to build a parallel interface for use with an A1 crate controller, similar to a former development for the PDP-11 minicomputer [5].

There are still many more possibilities, and it is evident that the personal computer can outgrow the state of a mere fancy toy to become a real "workhorse".

References

[1] A.J. Harding, The TRS-80 Owner's Report, Personal Comp. World, 1, no. 4 (Aug. 1978).

[2] ESONE Committee, Specificaton of the CAMAC Serial Highway and Serial Crate Controller Type L2, EUR 6100 e (1977).

[3] L.A. Van Rhee, in preparation.

[4] L.A. Van Rhee, Euromicro J. Special issue (September 1980).

[5] W. Stüber, CAMAC Bulletin 4 (1972).