Technical note

Microcontrollers as inexpensive pulse generators and parallel processors in electrophysiological experiments

D. Schild **A. Gennerich** **H. A. Schultens**

Physiologisches Institut, University of Göttingen, Humboldtallee 23, D37073 Göttingen, Germany

Keywords—*Microcontroller, Pulse generator*

Med. & Biol. Eng. & Comput., 1996, **34**, 305–307

1 Introduction

A SIGNIFIGANT PROPORTION of physiological experiments are usually carried out by applying a controlled stimulus to isolated cells or cells within a tissue and then measuring the response of the system. Often the specimen is placed in an electronic control circuit loop, such as in a patch-clamp experiment (HAMILL *et al.*, 1981). In most of these cases, the timing and amplitudes of the stimuli are predetermined by programmable stand-alone timer/stimulator units, or now more usually by the analogue and digital output from a computer interface. A problem arises when the computer has to handle a large amount of data at one time, e.g. video images or several channels of analogue input, writing these to disk at maximum speed. This bottle-neck can be obviated either by using two computers or by using one computer together with a preprogrammed timer/stimulator unit. However, communication between the two components of such a system can be slow and cumbersome.

A more elegant solution is to have two processor boards in one computer. We have previously developed a system for simultaneous patch-clamp and Ca^{2+} imaging measurements using a VME-Bus computer* with a Eurocom 5 main processor board and a SAC 700 secondary processor board (SCHILD and LISCHKA, 1994; SCHILD *et al;*., 1995).

In this paper we report on a similar but considerably cheaper method utilising a microcontroller card (MC) in a personal computer (PC).

Microcontrollers are now found in a rapidly expanding range of applications, but, to our knowledge, they have not yet been employed in electrophysiological experiments. As many experiments are currently using a PC, the additional costs for the microcontroller, some simple electronics and the necessary software are about one order of magnitude less than the costs for a second computer with the appropriate interface and software.

* Eltec, Darmstadt, Germany

First received 21 July 1995 and in final form 29 January 1996

2 Realisation of a PC/MC-based pulse generator

2.1 *Purpose and nomenclature*

A typical requirement in electrophysiological experiments is a sequence of analogue voltage steps and some digital outputs as triggers. We use four digital outputs and one analogue output at a resolution of 12 bits over a scale of -10 to $+10$ V. This combination is usually sufficient for patch-clamp experiments. Time resolution should be in the order of 1 ms. At any one moment, the output is defined as a fivetuple (U, T_1, T_2, T_3, T_4), where U = voltage and T_n = trigger (digital output) states (Fig. 1).

We call the period during which all parameters stay constant a segment. These are denoted S_1, S_2 etc., with voltages U_{S1}, U_{S2} etc. A pulse comprises a number of segments. The number of segments is limited only by the amount of dynamic memory available on the MC: each segment requires four bytes of MC memory, and the repeat interval extends these so that an MC with 16 KB of RAM can manage thousands of segments per pulse. In the example illustrated in Fig. 1*a*, the first pulse P_1 consists of four segments S_1, S_2, S_3 and S_4 with voltages U_{S1}, $U_{S2} = U_{S3}$ and $U_{S4} = U_{S1}$. The voltages, durations, and trigger states are programmable for each segment. Each trigger state can be set independently of the others to either 0 (low) or 1 (high) for the duration of a segment.

In addition, the voltage of a segment can be stepped automatically by a specified increment for successive pulses P_2, P_3 etc. In the example in Fig. 1*b*, which has the same segments as Fig. 1*a*, $U_{S1} = U_{S4}$ is held constant, and U_{S2} and U_{S4} are incremented by a positive value dU in each successive pulse. Similarly, the duration of a given segment can be increased or decreased by an additive or multiplicative factor in successive pulses. We call the train of successive pulses a pulse sequence. Fig 1*b* shows the voltage steps of a pulse sequence consisting of three pulses. The number of pulses in the sequence is determined by the initial and final voltages and voltage step size in the segment in which the voltage is varied. Pulse sequences can be repeated a desired number of times.

The above scheme, which we call a pulse protocol, is determined by the durations T_{S1}, $T_{S2}, \ldots$, of each segment; the initial voltage U_i, the final voltage U_e, and the voltage increment dU for each segment; the four trigger states in each

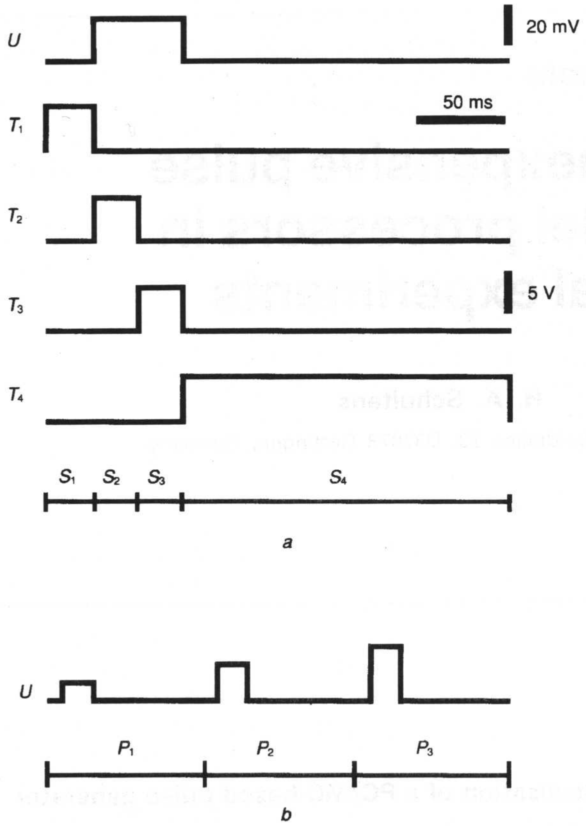

Fig 1 *(a) Simulated oscillograph traces (U, T_1–T_4) of the DAC voltage and the four trigger output levels, respectively; the time axis (S_1–S_4) indicates the segment intervals; $U_{S2} = U_{S3}$ and $U_{S4} = U_{S1}$ (b) pulse sequence with three pulses; the voltage in the second and third segments steps through three values*

segment; the number of segments per pulse; and the number of pulse sequences. The number of pulses per pulse sequence is calculated from U_i, U_e and dU.

2.2 *Microcontroller*

We use a microcontroller board (MC board) called PC-537-ADDIN, based on the 8-bit SAB 80C517A or 80C537 controller chips†§. This MC-board fits into an 8-bit slot in the back-plane of a PC. Basically, the MC-board has the following features:

(*a*) a controller with two programmable 16-bit timers/counters and 256 byte on-chip RAM; furthermore, the SAB 80C517A additionally contains 2 KB of on-chip RAM (called XRAM).

(*b*) external program and data memory expandable up to 64 KB; less than 16 KB are needed for our application.

(*c*) a port for communication with the PC via the ISA bus.

(*d*) 24-bit addressable digital inputs/outputs, 16 of which we use.

Technical details of the microcontroller and the hardware can be found elsewhere (FEGER and ORTMANN, 1994; SIEMENS, 1994).

Programs that run on the MC can be developed on the PC and downloaded into the MC via the PC bus. After loading the program into the MC, the parameter set that defines a pulse protocol can be passed to the MC. The program downloaded into the MC can be started by the user routine running on the PC. This allows interactive control of the MC pulse program. Fig. 2 shows a sketch of the PC, the MC and the external board (see Section 2.5).

2.3 *Pulse program editor*

The timing of triggers and segment voltages of a pulse protocol are edited by the user. We have written a program that gives the user the menus shown in Tables 1 and 2. It allows the operator to choose all parameters of a pulse sequence. Most items in the menus are self-explanatory. Two MC boards can be handled by this program.

Some of the segments in a pulse can be repeated by providing the parameters S_m, S_n (repeat segment from, to) and R, the repeat count. This causes segments $S_m - S_n$ to be repeated the specified number of times before going on with segment S_{n+1}. This feature is useful, for example, when a certain number of action potentials are to be generated at a desired rate before a specific current or voltage response is to be measured. Further, the parameters dT and xT allow an additive or multiplicative increment of the duration of a segment. This is useful when studying the inactivation or recovery from inactivation of membrane conductances.

Finally, the contraction factor c shortens all segment durations by this factor. This is a convenience that allows otherwise very long pulse sequences to be tested by observing them on an oscilloscope. A graphical representation of pulse sequences on the computer screen is therefore unnecessary, but

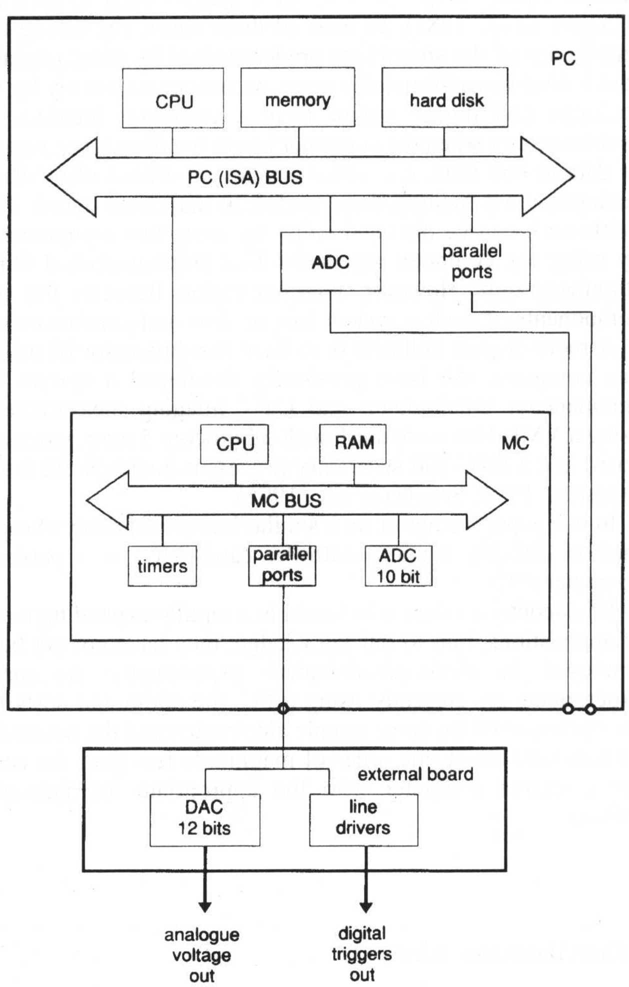

Fig 2 *Microcontroller-based pulse generator as a PC add-on system*

† Siemens 8051 8-bit microcontroller family

§ Feger & Co., Traunstein, Germany

Table 1 Main menu of pulse editor/loader program

Controller status		1: running	2: (waiting)
<F1>	Halt/Restart controller 1		
<F2>	Halt/Restart controller 2		
<F3>	Halt/Restart controller 1 and 2		
<F10>	Boot microcontrollers		
d:	Directory for parameter files:	C:\user\AG\pls	
l:	Load parameter file	seal.pls	
s:	Save parameter file	Na.pls	
e:	Edit parameters		
p:	Put parameters into controller		
g:	Get parameters from controller		
#:	Show parameter file		
^L	Refresh screen		
X,^C	Exit (without or with stopping MC program)		

Table 2 Pulse edit menu

Editing parameter file:		seal.pls
f:	Number of pulse sequences	10000
n:	Number of segments	5
r:	Repeat segment from, to	3,4
R:	Repeat count	10
o:	U offset	0
s:	U scale	0·05
c:	contraction factor	1
Segment #:1		
u:	voltage Ustart, Uend (mV)	0,160
d:	voltage increment dU (mV)	20
t:	Duration Tstart, Tend (ms)	20
+:	dt	0
*:	xt multiply factor	1
T:	Triggers	0010
<PgUp>	Previous segment	
<PgDn>	Next segment	
<Esc>	Return to main menu	

they can be printed out as a list of voltage values and times for each segment.

The edited parameters of a pulse program can be passed to the MC or they can be stored on the hard disk for later loading. In this way, a library of pulse programs can be generated. When the parameters of a specific pulse protocol are passed to the MC, the MC program (which has previously been downloaded into the MC, usually when the PC is booted) can be started. In the present version its functions are to

(*a*) take the duration of segment 1, load it into the counter register of a timer in the MC and start the timer at the appropriate clock frequency (the counter register is incremented every instruction cycle, 1 μS instruction cycle at 12 MHz oscillator frequency); the MC has its own crystal-controlled clock, and so the segment duration is precisely timed and independent of PC timing.

(*b*) put the high eight bits of the segment voltage out on port A and the low four bits out in the low nibble of port B; the four trigger bits of the segment are put into the high nibble of output port B.

(*c*) wait until the timer started in (*a*) gives an overflow (the overflow flag can be used to request an interrupt), indicating that the first segment is over.

(*d*) repeat the steps (*a–c*) for all segments of the first pulse sequence.

(*e*) repeat steps (*a–d*) for the required number of pulse sequences.

In our application, the 16 bits of port A and B are the output of the MC. The lower 12 bits are converted to an analogue signal, and the remaining four bits are used as digital trigger signals. The MC processor is sufficiently fast that the order of the steps (*a*) and (*b*) above does not affect our experiments. It takes about 4 µs, which is comparable to the rise times of the amplifiers involved and negligible compared to the rates of the processes to be measured.

2.5 *DA convertor*

As noise is of crucial importance in electrophysiological experiments, we connect the 16-bit output of the MC to an external board made for mounting in a 19 inch rack, rather than mounting the DAC on the MC board. The external board simply consists of a 12-bit DAC** with ±10 V buffered output and a line driver for the four trigger signals.

3 Discussion

The microcontroller-based pulse generator described in this paper has several advantages over most other types of pulse generators.

In most laboratories there are PCs from XT to AT 286s, which are no longer used due to the hardware requirements of newer software packages. These computers can easily be equipped with an MC board and converted into programmable pulse generators. The necessary software, the pulse program editor and the MC pulse program are easy to develop, and can be obtained from the authors. We used Borland Turbo C++ for the PC editor program and a C cross-compiler (C51 compiler)†† for the MC programs.

This application is particularly useful when the MC is incorporated in the PC performing the data recording, such as DMA sampling of electrical signals or image acquisition.

The MC/PC combination is a very general system that can be used for many other types of experiment in the electrophysiology laboratory. Both components are freely programmable and quickly adaptable to many problems once the technique has been developed for one experiment. For example, the MC board that we use has a 10-bit ADC on it that would allow the MC to sample additional data or ambient parameters and control the experiment interactively. Furthermore, digital inputs to the MC board could be used as signals to the program running in the MC quite independently of the PC program.

In some experiments, it is necessary to have two amplifiers when two-cells are recorded simultaneously. For these experiments, two microcontroller boards can be placed in the same PC to run two different pulse programs simultaneously in a co-ordinated way.

References

FEGER, O., and ORTMANN J. (1994): 'MC Tools 14'. Feger + Co. Hardware Verlag, Traunstein, Germany

HAMILL, O. P., MARTY, E., NEHER, E., SAKMANN, B., and SIGWORTH, F. J. (1981): 'Improved patch-clamp techniques for high resolution current recording from cells and cell-free membrane patches', *Pflüger's Arch.*, **391,** pp. 85–100

SCHILD, D., and LISCHKA, F. W. (1994): 'Amiloride-insensitive cation conductance in *Xenopus laevis* olfactory neurons: a combined patch clamp and calcium imaging analysis', *Biophys. J.*, **66,** pp. 299–304

SCHILD, D., LISCHKA, F. W., and RESTREPO, D. (1995): 'InsP$_3$ causes an increase in apical $[Ca^{2+}]_i$ by activating two distinct current components in vertebrate olfactory receptor cells, *J Neurophysiol.*, **73,** pp. 862–866

SIEMENS (1994): 'Microcomputer components, SAB 80C517/80C537, 8-Bit CMOS' Siemens User's Manual 08.93

** DAC80, Burr-Brown Corporation

†† Keil Elektronik GmbH, Grasbrunn, Germany