Low cost construction kit:

REACTION TIMER BASED ON A MICROCONTROLLER

Here's a modern high-tech version of the familiar 'first to press their button' game. Thanks to the use of a low cost pre-programmed microcontroller chip, it's not only very easy to build but also displays the actual reaction times in milliseconds, of the 'winner' and 'loser'.

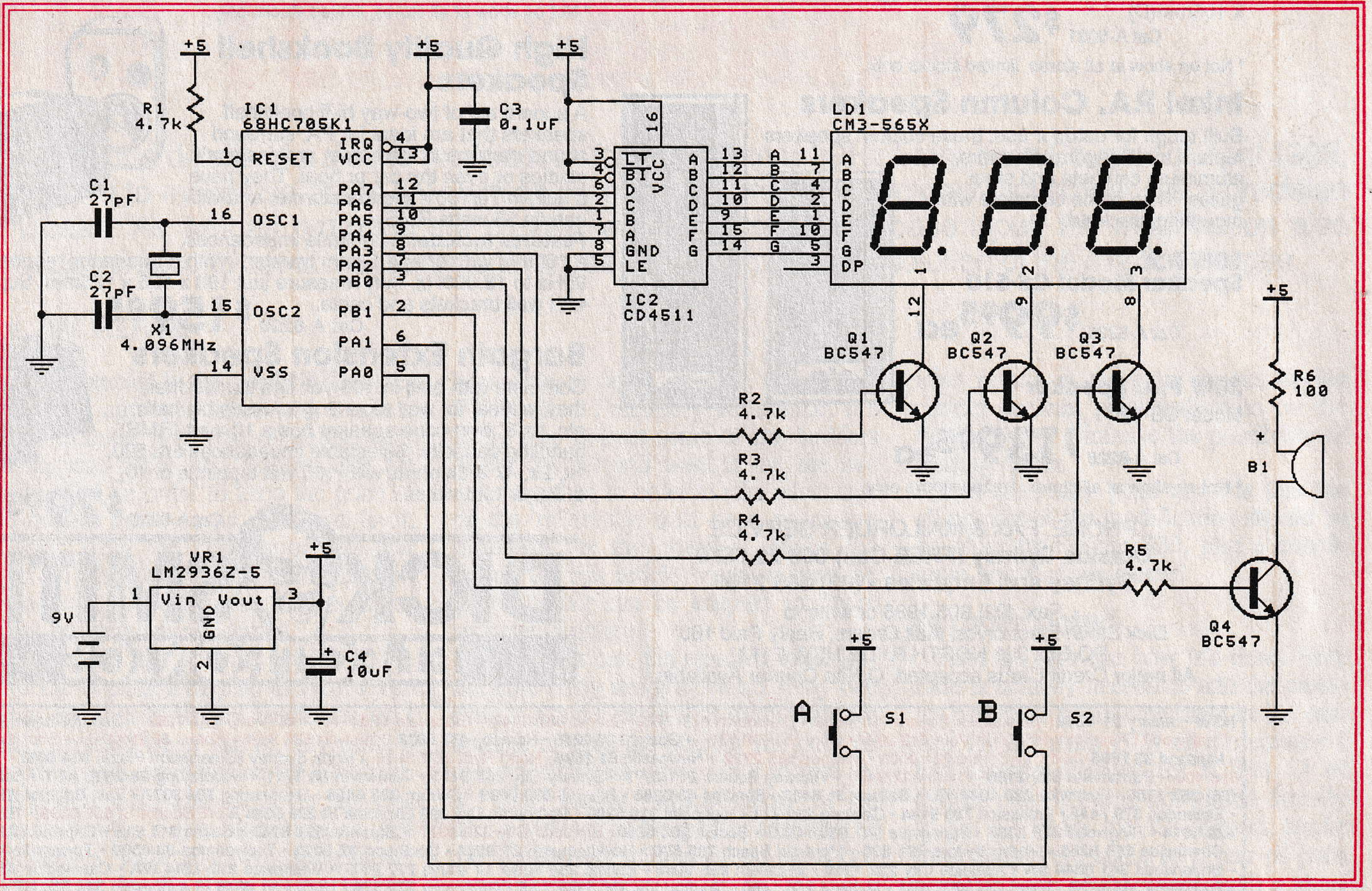
by PETER CROWCROFT

A quiet revolution is taking place in electronics today, one that is almost as important as the replacement of valves by transistors and ICs in the 1960's and 70's. This is the rise of the microcontroller. (For convenience we will often abbreviate microcontroller to 'uC' in the text which follows.) The increasing power and versatility of these devices, coupled with steadily decreasing prices, is now such that many of the simplest logic circuits may now be better designed using them.

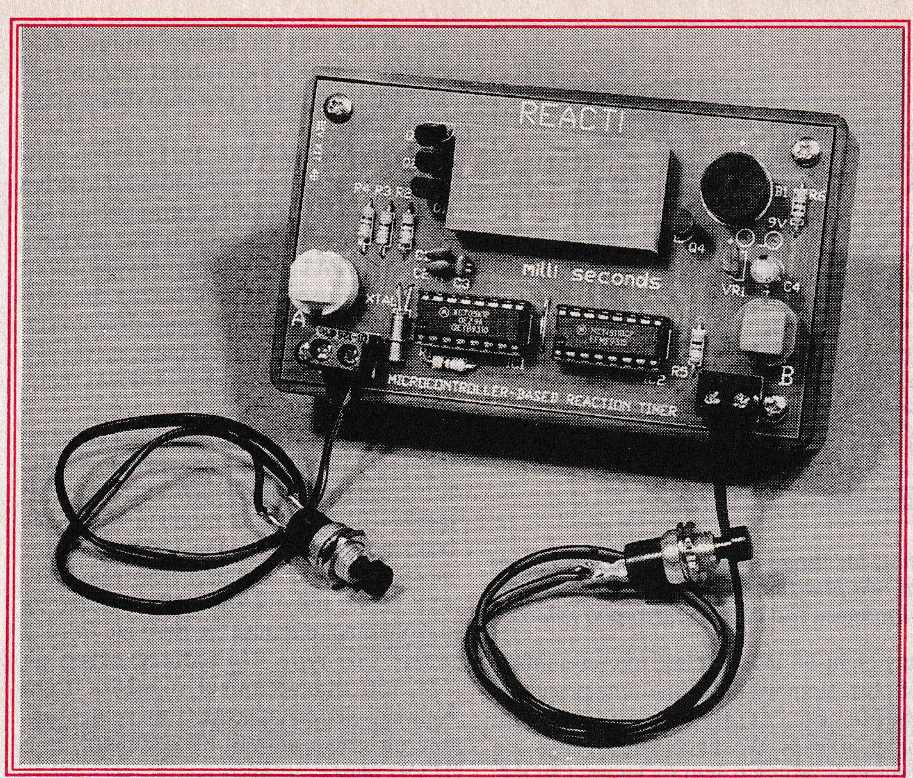
Microcontrollers are basically a complete computer on a single chip — a central processing unit plus ROM and RAM memory, a clock, input/output ports and often inbuilt peripheral devices (such as a timer, AD/DA and communication ports).

Today they are to be found all around us. To name a few applications, they're now found in washing machines, microwave ovens, mobile phones, bar code readers, printers, video cassette players, photocopiers and, most importantly, motor cars — for engine fuel management, braking systems, active suspension, multiplexed wiring, mirror and window movement.

In all of these cases the application program inside the ROM of the uC is not accessible to the hobbyist and therefore of



The schematic for the reaction timer. As you can see the 68HC705K1 microcontroller chip IC1 does most of the work, under the control of its built-in firmware. As a result very little external circuitry is required.



little use. Even if it was, the tools to develop programs for these devices are expensive — typically over US$1000.

It has generally also been true that to learn to program uC's has required attendance at a technical college or some similar formal teaching institution. To try to teach yourself programming of the older uC's, such as the 8748 or 8051 family, from a data book was not easy.

Times have changed

This situation has changed in the last year or two. Low cost uC's and the tools to develop and program code for them have now reached the level where the amateur can become involved.

In this respect two chips have led the way. They are the PIC series from Microchip and the K series chips from Motorola. The PIC chips have been out longer, but the development tools to use them are relatively expensive (though coming down) and the instruction set is a bit strange.

We will not discuss them further here. (Small uC's from other manufacturers also suffer from the problem of high cost of tools for code development and programming, even higher than the PIC uC's.)

The beauty of using the K1 Motorola range of uC's is that the full range of development tools needed by the amateur to develop applications are readily available at low cost, say under US$200. The programs (called assemblers) to develop code for the uC on a PC are freeware, and are available from Motorola and its distributors. The hardware schematic for the programmer is published in the Motorola Data Book, and the software to use the programmer is also freeware. A low-cost in-circuit simulator, which also acts as as programmer, is available also with software, from Motorola dealers.

The K series uC chips come in various packages, including the EPROM version. A truly excellent manual which introduces the K series from basic principles, called *Understanding Small Microcontrollers* is also available from Motorola.

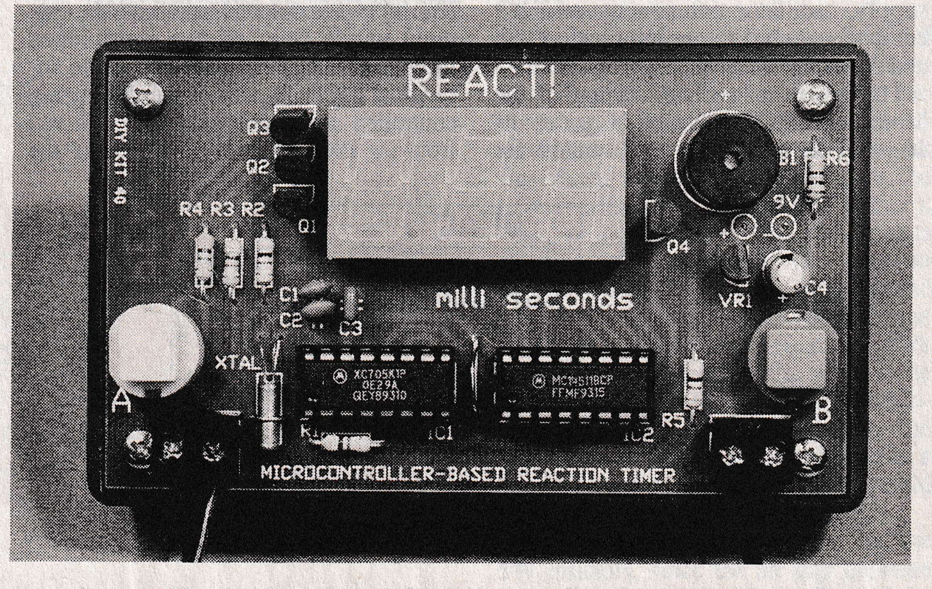
This project & kit

This reaction timer project is designed to provide a low cost application of the 68HC705K1 uC, to demonstrate how easy it is to use this kind of chip for simple real-world tasks. Using a microcontroller in a reaction timer is cheaper, uses fewer components and allows a flexibility not possible with straight logic components.

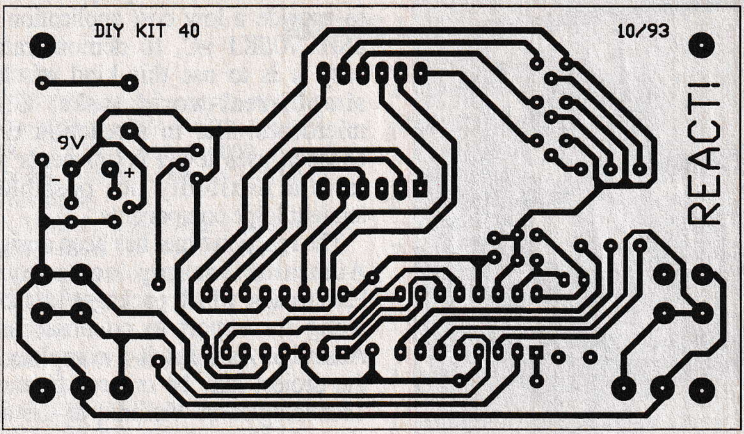
The reaction timer has been designed in Australia for my company DIY Electronics, which packages kits in Hong Kong for export to countries such as Australia. As a result a complete kit for the project may be ordered from Alpine Technologies in Victoria, at the address shown in the parts list. The reaction timer kit costs $45.90 and may be paid for by cheque, money order, Mastercard or Bankcard. The kit will be posted to you directly from Hong Kong, and any import or other duties which may apply must be payable by the buyer (if required); they are not included in the purchase price.

Either one or two players may use the timer. Their reaction time to press a button, after a 'beep' sounds, is accurately measured to the nearest thousanth of a second and displayed on the LED display. If two players play, then both reaction times are displayed in turn. The beep occurs after a random delay of anything between zero and four seconds. The system uses a 4.096MHz crystal for accurate time measurement.

All the software code to operate the timer is squeezed into just 1EA hex (490 in decimal) bytes of ROM, which is almost all the ROM space available inside the K1 (1F0 hex or 496 decimal). The complete source code is provided



A close up of the top of the reaction timer, showing where everything goes — apart from the battery, which mounts inside the box underneath.



Here is the PCB pattern for the reaction timer, for those determined to etch their own. Although the pattern is not copyright, we imagine that most builders will want to use a DIY Electronics kit — which includes the preprogrammed micro.

on a floppy disk provided with the kit, fully commented so that you can use it to teach yourself 68HC705K1 programming. This is actually the second kit in a planned series designed to show how microcontrollers are used and programmed. The first kit is a down counter (see box).

The reaction timer kit is constructed on a single-sided printed circuit board, which fits into a small plastic jiffy box. Protel Autotrax and Schematic CAD packages were used to design it. The firmware for the 68HC705K1 was developed using a Motorola ICS (in-circuit simulator), purchased from Motorola.

Circuit description

The heart of the circuit is the 68HC705K1 uC (IC1) and the firmware it contains. The other items are the power supply, oscillator components and I/O components used to get information into and out of the uC.

The K1 is a low-end version of the Motorola 68HC05 eight-bit uC series. Its 16-pin DIP package is the smallest for any eight-bit uC, with the lowest number of pins. Here we use a one-time programmable or 'OTP' version of the 68HC705K1. We have already put the firmware program into its internal ROM, and since there is no quartz window in it, the program cannot be erased.

The K1 chip has its own on-chip clock oscillator, and here we have used a 4.096MHz crystal (X1) with capacitors C1 and C2, to make it run at an accurately defined frequency. Successive division by two in the 15-bit multi-function timer inside the K1 generates a TOF (timer overflow) interrupt every 500 microseconds. Every second interrupt is used to increment a 1ms counter inside the chip, and this becomes the actual reaction timer.

IC2 is a BCD to seven-segment decoder/driver, which is used to take BCD data from I/O lines PA4-7 of the K1 (configured as outputs), and drive the segment anode lines of the multiplexed three-digit LED display LD1. The individual digit cathodes of the display are controlled using transistors Q1-3, driven from the PA3, PA2 and PB0 lines of the K1. In this way the display multiplexing is done using a minimum amount of hardware, with the firmware doing most of the work.

Another K1 I/O line, PB1, is also configured as an output line and used to drive transistor Q4, which in turn controls the piezo beeper B1. The remaining I/O lines PA0 and PA1 are configured as input lines, and used to connect to the two pushbuttons A and B. No pull-down resistors are required on the inputs, as the K1 provides programmable pull-down resistors on chip.

When +5V is applied to either of these inputs by pressing either pushbutton, the K1 'awakens' from its low-power STOP mode. (It enters this mode if there has been no activity on the input lines for more than one minute.) The regulated +5V power rail needed by the K1 and its surrounding circuitry is derived from a small 216-type 9V battery using an LM2936 three-terminal regulator from National Semiconductor. This is an ultra-low quiescent current 5V regulator, in a TO-92 plastic package.

There is no hardware off/on switch, as the current drain from the battery when the K1 uC is in STOP mode is so low. We measured 25uA battery drain when the K1 was in STOP mode, and 10mA when it was running.

Firmware description

The uC sits in a powered-down state (STOP mode), drawing only a few microamps until either key is pressed. The timer is based around the multi-function timer in the K1. The Real Time Interrupt (RTI) generates an interrupt every 16ms (milliseconds), and this is used to:

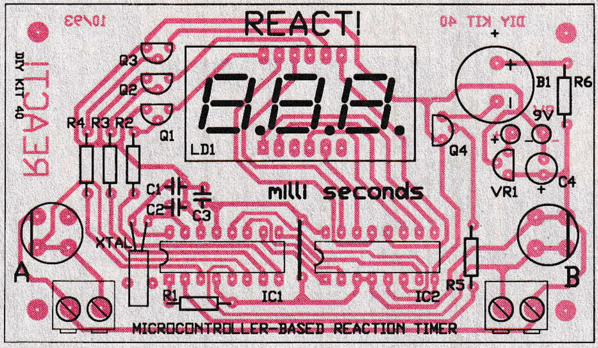
- Count the time since the last key was pressed, so that the unit will re-enter STOP mode after one minute of inactivity;
- Scan the display and show the results.

The TOF interrupt generates an interrupt every 0.5ms, as noted earlier. This is used to:

- Do 'beeps';
- Count the reaction time;
- Check for key presses.

To see how the firmware works, print out the **REACT.LST** program contained on the floppy disk which comes with the kit. This is a combined listing of the program code, the mnemonics (assembler language) used to write it, line numbers for reference — and, most importantly, a lot of comments to explain what is happening.

Along with the listing file, the disk also provides the assembler source file **REACT.ASM** and a header file **K1DEF.ASM**. The purpose of the latter



And finally, here's the overlay diagram. Use this together with the close-up photo, as a guide when you're wiring up the project.

is to provide the assembler with definitions of the mnemonics used in the source file itself.

It is difficult to remember binary bit patterns and hexadecimal addresses when you write code. It is far easier to remember mnemonic names for register addresses and bit positions. Using these names also makes the program easier to read. **K1DEF.ASM** consists entirely of EQUate directives, defining these mnemonics as labels for a binary number or address. Note that this is a general purpose header file. Not all labels defined in the header file are used in *REACT.ASM*. You are free to use *K1DEF.ASM* when you write your own K1 programs, so print it out and examine it. **K1DEF.ASM** is called by **REACT.ASM** using the *$include* directive, when timeout is assembled.

As well as providing the full firmware files on floppy disk, the kit also provides printed flowcharts to make its operation clearer. There are a number of flowcharts, as the firmware may be divided into several functional sections. The K1 memory map is also provided.

(Note that none of the program listings or flowcharts are provided in this article, as they would take up far too much space. However if individual readers send a formatted (MS-DOS) 1.2MB or 1.4MB floppy disk to the *EA* Reader Information Service, with the usual fee of $7.50 to cover handling and postage, we can provide a copy of all files.)

Assembling the kit

Check the components against the parts list (also provided as a file on the floppy disk). Place them according to the overlay diagram. Make sure to get the buzzer, 'hatkey' pushbutton switches and capacitors around the correct way. The LED display unit must have the decimal points on the bottom. There is no off/on switch.

There is a single wire link to be fitted to the board, between the two ICs. The ICs also fit in sockets, and needn't be fitted until the rest of the circuit is completed. Tie-down pads have been provided for a loop-link to secure the crystal to the PCB. The crystal can be inserted either way around.

Note that you have a choice of using either the 'hatkey' switches which mount directly on the PCB, or the off-board switches which connect to the two small terminal blocks.

The battery stays inside the box, so solder the wires of the battery snap to the copper side of the board. Stick the battery down inside the box with some adhesive or double-sided tape. It will last for many months of use, because of the ultra-low current voltage regulator used in the circuit. (Our prototypes are still working very well on their original batteries, despite hours of testing.)

Using it

Pressing a switch turns the unit on. (There is no separate off/on switch, remember; the activity level is all done in firmware.) The display normally shows the fastest reaction time in milliseconds (thousandths of a second), achieved so far by any player. This time is accurate to over two decimal places. However the very first time the timer is used after the battery is connected to it, there is no fastest time to display, so a '999' is displayed...

Press either key to trigger the unit. The display will change to 000. After a random period of time between zero and four seconds, the unit will beep and the display will clear. Now press either button as quickly as you can. The display then shows '1' (to indicate the winner), in the display digit closest to the key that was pressed first. Then it will show the winner's reaction time in milliseconds.

The display will then show the time taken in milliseconds for the second key to be pressed — if it was pressed. The display will cycle through the times and first/second indicators until a key is pressed to return to the start. Only reaction times to 999ms are recorded. If you cannot react to a beep in that time, then perhaps you better give up electronics and try lawn bowls...

If a key is pressed *before* the beep, then no time is displayed for that key. The other key, if pressed legally will display the reaction time as usual. If no keys are pressed within a second, '999' will be displayed.

Should both keys be pushed at the same time, only the one time will be displayed. If no keys are pressed for one minute the unit will switch itself off automatically. Terminal blocks, twin cable wire and push-on switches are included with the kit, so you can use the timer at a comfortable distance away from it.

Final comments

If your reaction timer doesn't work, poor soldering is the most likely reason — so check all of your solder joints carefully under a good light. Next check that all components are in their correct position on the PCB, especially the two ICs, the transistors and capacitors.

If the display turns on *very* brightly, remove the battery because the oscillator circuit is not working — and there are no current limiting resistors to protect the display (the brightness is all controlled by the firmware). Make sure you did not mix up the voltage regulator with one of the transistors — they both come in similar TO-92 packages.

Also make sure you added the link to the PCB, and check that the flats on the katkey switches are facing outwards, as indicated on the overlay...

Finally, for suggestions on learning 68705 assembler code, see the **READ.ME** file on the floppy disk. ❖

'TIMEOUT': another DIY kit

In addition to the reaction timer kit described in this article, Dr Crowcroft's company DIY Electronics has developed another small kit to provide further 'hands on' experience with programming and use of the Motorola 68HC705K1 microcontroller.

Dubbed 'Timeout' and carrying the number K38, the kit is for a simple down-counter which counts for either 60 or 90 seconds, beeping every 10 seconds — until the final 10 seconds, when it beeps every second.

This kit is also based on the 'K1 microcontroller, and has a very similar physical arrangement as the reaction timer except that there are only two single-digit LED displays. As with the reaction timer, all firmware source code is provided on floppy disk as well as flowcharts, schematic, PCB pattern, etc in the documentation. The K38 kit is available for $39.50, in the same way as the reaction timer kit.

PARTS LIST

Resistors

1/4W carbon, 5%:
1 100 ohms (R6)
5 4.7k (R1-5)

Capacitors

2 27pF ceramic (C1, C2)
1 0.1uF monolithic (C3)
1 10uF electrolytic (C4)

Semiconductors

1 68HC705K1 OTP microcontroller (IC1)
1 CD14511 BCD/7-seg decoder-driver (IC2)
1 LM2936-5 voltage regulator (VR1)
4 BC547 NPN transistors (Q1-4)
1 CM3-5655 three-digit LED display

Miscellaneous

1 PC board, 96 x 55mm, coded K40
1 Plastic case, 100 x 60 x 30mm, with screws
2 Push-on 'hatkey' switches
2 Two-way terminal blocks, PCB mounting
1 Piezo buzzer, 5V (B1)
1 4.096MHz quartz crystal
9V battery snap lead; two small push-on switches; six- foot length of twin-lead cable; two 16-pin DIL sockets for ICs; documentation and floppy disk.

Note that all of the above parts are available in a kit priced at $45.90. Orders should be sent to Alpine Technologies, PO Box 934, Mt. Waverley 3149. Kits will be despatched from Hong Kong via airmail.