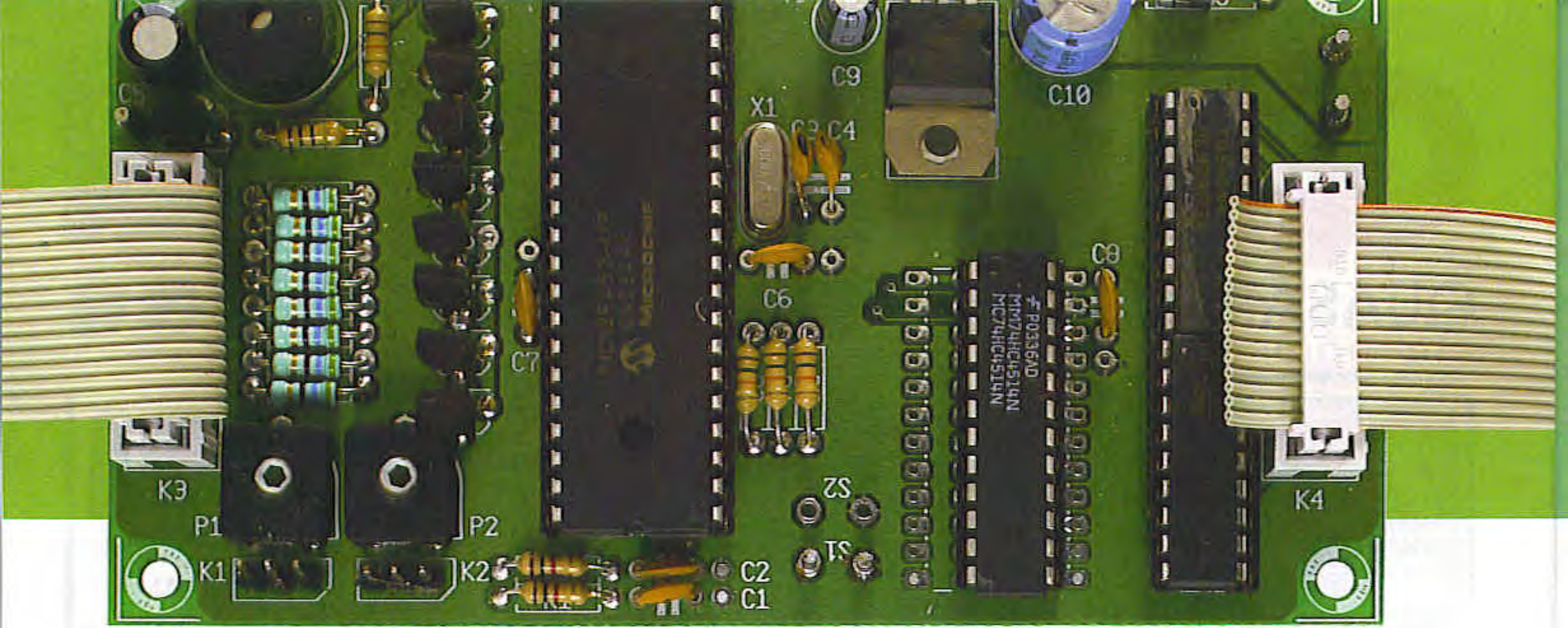
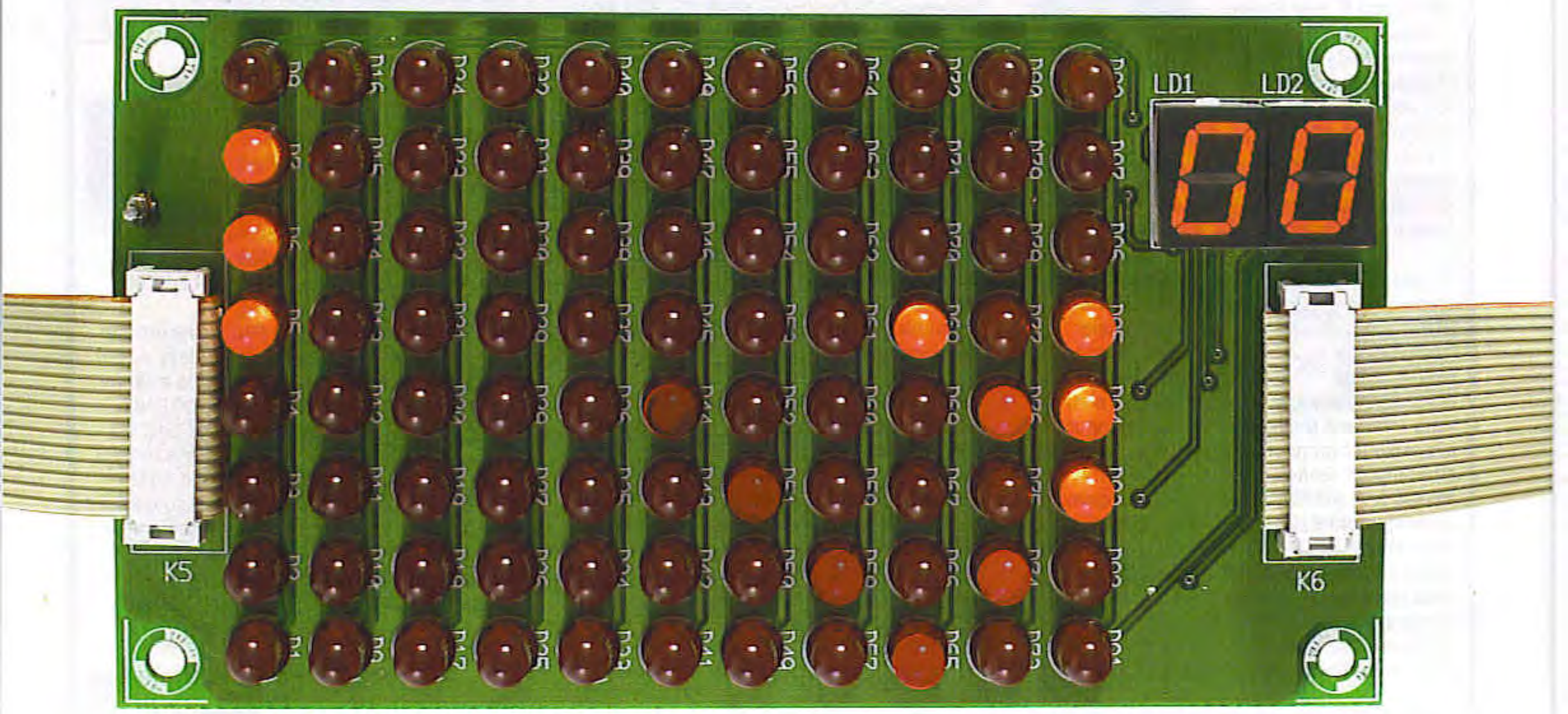


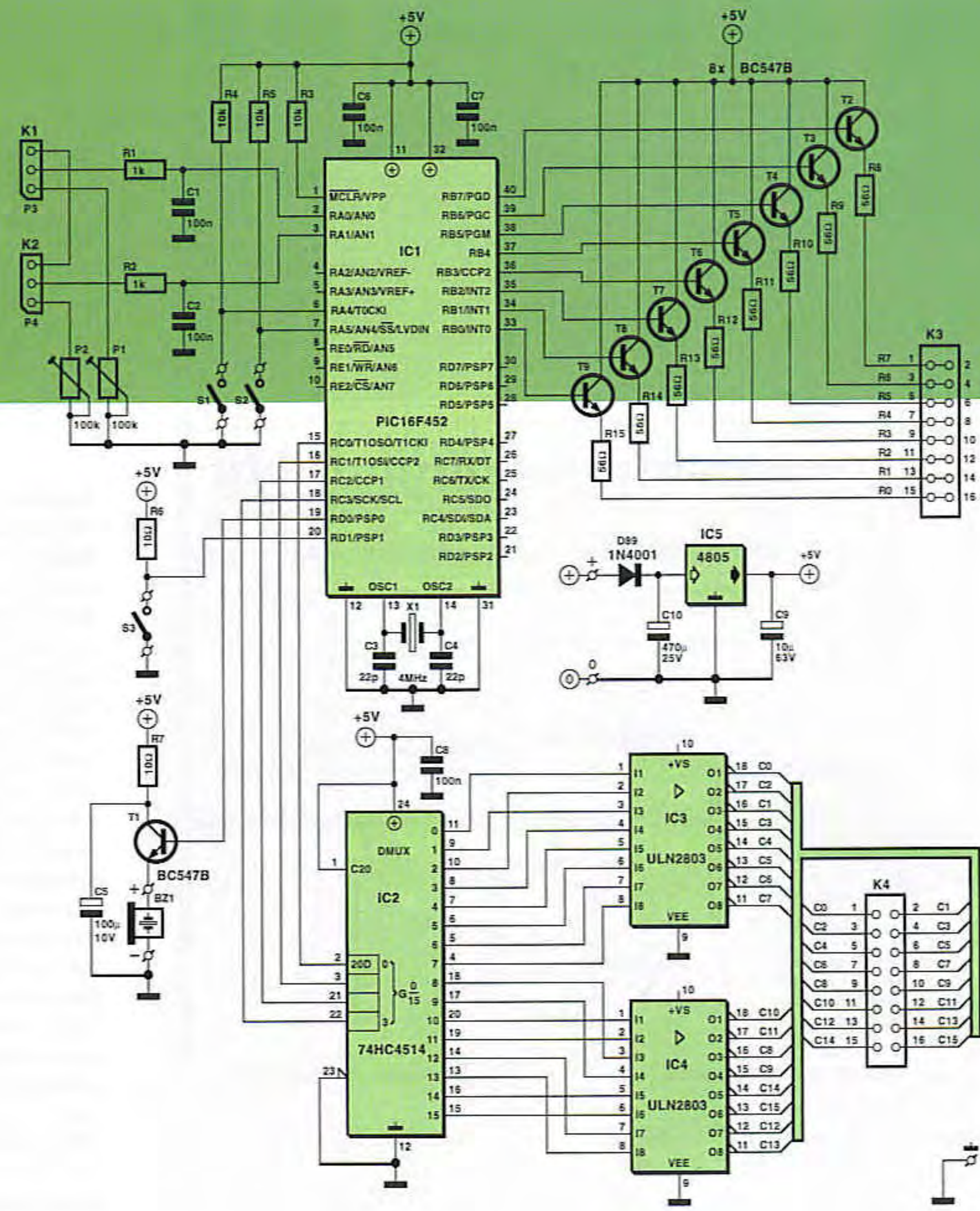
Pocket Pong

a primeval game cast in modern hardware



Andy Morell

Provided they manage to recognise them in the first place, youngsters will label classics like Pacman and Pong as video games although historically they are 'video games', the concept behind them dating back to the 1950's. In this article you'll find a modern (computer) version of such a prehistoric game that — as far as we are concerned — has not lost any of its compulsive character.



It is easily forgotten that the first electronic games were played on a TV set. In fact, technology at the time nearly did not make it to TV altogether. In 1951, TV technician Ralph Baer thought it would be nice to use the telly as a screen for an electronic game. His boss however did not see the promise and the idea was quickly abandoned. Years later, however, it started to surface again and in 1966 Baer started to build one of his early prototypes. The video game was born. The game covered by this article is Atari's 'Pong' which is actually a derivate of one of Baer's original concepts. The first versions for use at home were designed around 1974. As opposed to other manufacturers, Atari found the pot of gold: an ASIC (application specific integrated circuit) was designed for Pong. The chip allowed the production costs to be kept low while the game functionality (including a digital on-screen scoreboard and sound effects) was excellent compared to competitive products. The home version of Pong was launched in 1976. Today, 28 years later, we have another go at casting Pong in electronics. This time, we will not be using a TV set for the 'screen' but a LED matrix.

The circuit

Just like Atari did many years ago, we will be designing a chip tailored to the game only. Fortunately, that no longer means you have to design a completely new circuit and burn it into a chip. Today we simply use a microcontroller running software that tells it exactly what to do. We chose the PIC18F452, a 40-pin MCU containing, among others, 32 kBytes of program memory and a 10-bit A-D converter. As you can see from **Figure 1**, the PIC is not the only IC in the circuit. IC2, a 4-to-16 line decoder, together with IC3 and IC4 arranges the display control. The display actually consists of two parts: the 7-segment displays LD1 and LD2 showing the 'score' and a LED matrix (D1-D88) that mimics the playing field. Virtual rackets or bats move at the left and right side of the court, allowing the ball to be bounced back and forth.

Buzzer Bz1 provides the sound effects. You are looking at a dc (or 'active') piezo buzzer that's driven by transistor T1. C5 and R7 afford sufficient decoupling of the supply voltage.

The power supply around IC5 is dead standard. Diode D89 affords a degree of protection against an accidentally

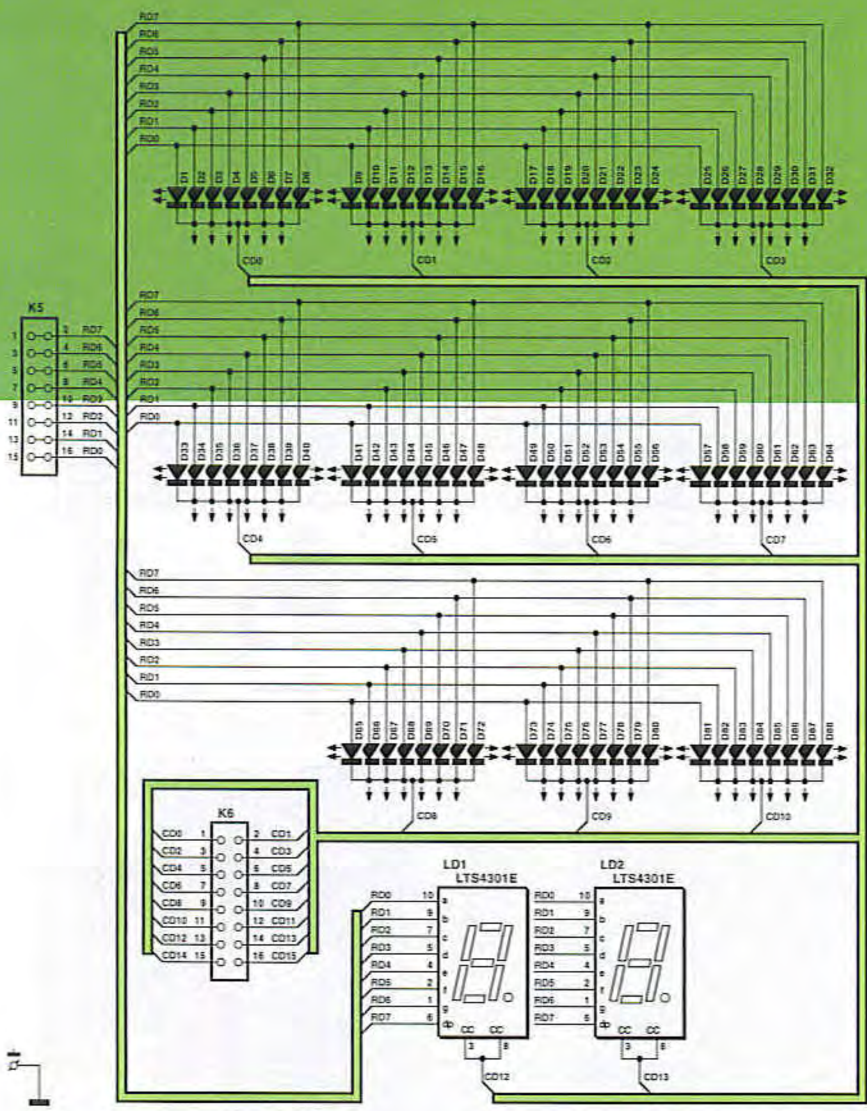


Figure 1. In this game, the 'screen' is formed by a huge number of LEDs.

COMPONENTS LIST

Resistors:
R1,R2 = 1kΩ
R3-R6 = 10kΩ
R7 = 10Ω
R8-R15 = 56Ω
P1,P2 = 100kΩ preset

Capacitors:
C1,C2 = 100nF
C3,C4 = 22pF
C5 = 100µF 10V radial

reverse polarized mains adapter (with 9-12 VDC output). With the PIC drawing just a few milliamps, it is fair to say that the current consumption of our electronic game goes on account on the LEDs. However, thanks to the multiplexed drive scheme used here, the average current consumption remains limited to a modest 35 mA or so.

Display

Both the LEDs in the 7-segment displays and the LEDs in the matrix have their cathodes connected via ULN2803 driver ICs (IC3 and IC4). The anodes are connect to the MCU port lines via transistors T2-T9. It would appear that the transistors are not strictly necessary as the PIC port lines are specified at 25 mA each. This may well be sufficient for high-efficiency LEDs, but it isn't with regular LEDs which at such a low current light dimly, reducing the 'playability' of the game.

Using the indicated component values (i.e. with transistors and R8-R15 = 56 ohm) a LED current of about 27 mA is obtained. By the way, the value of R8-R15 may be changed without problems using Ohm's law. Assume a supply voltage of 5 V, then subtract the following: collector-emitter drop (0.7 V); LED 'on' voltage (approx. 1.8 V for red LEDs); voltage drop across Darlington drivers in the ULN2803 (approx. 1 V). That leaves about 1.5 V across the resistor. If the desired current is 10 mA, $V = I \times R$ tells you that $1.5 = 0.01 \times R$, or $1.5 / 0.01 = 150$ ohms.

Operation

The game is played using two potentiometers and two switches. S1 serves to serve a ball. S3 is the speed selector. When it is closed, the ball moves faster making the game more difficult to play.

There is a connection for a third switch (S2) but this is not used in the Pong game. P3 and P4 may be ordinary rotary potentiometers but slide pots will of course give a more realistic control of the bats on the field. A real joystick is of coursed the ultimate.

'Analogue' PC joysticks in general contain two potentiometers, one for each direction (horizontal and vertical). In most cases 470-kohm pots are used of which the 0-120 kohm resistance range is actually used. For our circuit, a resistance range of 0-4.7 kohm is required, so if a joystick is connected, a resistor has to be connected in parallel with the input (between +5 V and pin 2 / 3 of the PIC) to make sure a much lower resistance is obtained. The equivalent resistance of the parallel network is calculated from

$$1/R_{eq} = 1/R1 + 1/R2$$

So, if we want 4.7 kohms and the joystick R1 = 120 kohms then

$$1/R2 = 1/120k - 1/4.7k$$
$$R2 = 4.89\ k$$

In practice no problems will occur if you use a resistor of 4.7 kohms.

Pins 1 and 3 on the 15-way joystick connector (a sub-D type) are for the horizontal direction and pins 1 and 6 for the vertical direction.

Finally, R1/C1 and R2/C2 suppress noise generated by the potentiometers as they are operated.

Construction

The printed circuit board (**Figure 2**) consists of two parts interconnected with a piece of flatcable. Building the LED matrix is sure to take some time. Although the job itself is straightforward, we should emphasize the importance of checking the LED polarity because it is hard to think of anything

C6,C7,C8 = 100nF
C9 = 10µF 63V radial
C10 = 470µF 25V radial

Semiconductors:
D1-D88 = LED, red, high-efficiency, 5mm, e.g., HP HLMP-D101 (Farnell # 323-044)
D89 = 1N4001
LD1,LD2 = LTS4301E LiteOn
T1-T9 = BC547B
IC1 = PIC18F452-I/P
IC2 = 74HC4514 (74HCT4514 or 4514)
IC3,IC4 = ULN2803
IC5 = 4805

Miscellaneous:
K1 = P3 = 4kΩ7 (5kΩ) linear potentiometer, and a 3-way SIL pinheader
K2 = P4 = 4k7 linear potentiometer, mono, + 3-way SIL pinheader
K3,K4 = 16-way boxheader
K5,K6 = 16-way flatcable connector for PCB mounting
S1 = pushbutton, 1 make contact
S2 = not fitted
S3 = switch, on/off
X1 = 4MHz quartz crystal
BZ1 = 5V (DC) piezo buzzer

more annoying than 88 LEDs fitted the wrong way around. Usually, the cathode is the largest surface inside the LED as well as the shorter pin. Usually… not always, so make sure of the polarity and in case of doubt use a conductance tester.

The orientation of the 7-segment displays may appear to be wrong but if you follow the indications on the component overlay shown in Figure 2 the circuit will work as expected.

The polarity of the electrolytic capacitors and transistors in the circuit also deserve your attention. Also, run a double check on the orientation of the PIC micro before inserting it into its socket — after all, the PIC is the most expensive component.

The circuit board is best mounted into a case that will also accommodate the switches, mains adapter socket, pots and LEDs. A red bezel on top of the matrix clearance and the score displays will provide the finishing touch to the game.

If necessary the game may be powered by four AA batteries. In that case the current through the LEDs has to be reduced, however, by increasing the value of R8-R15. This is necessary to save battery power.

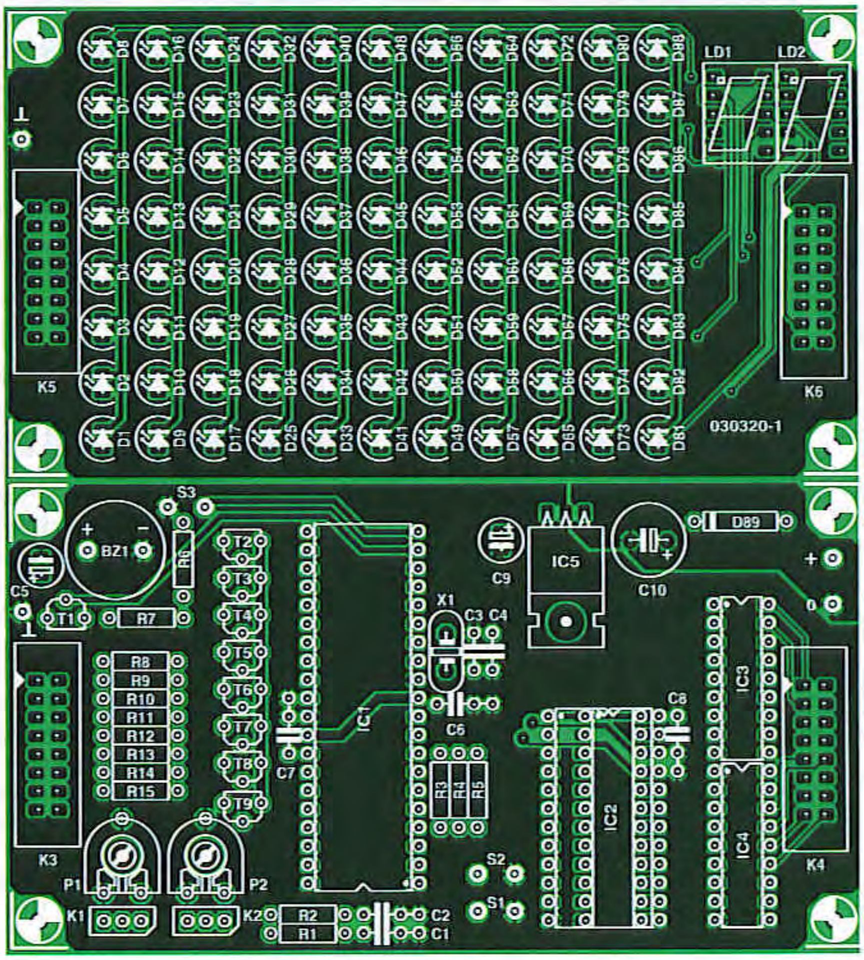


Figure 2. The PCB consists of two parts to be connected with flatcable.

Software

That leaves us with the software burned into the PIC micro, here, a PIC18F452. Since we are looking at a game that should not cost too much, we looked for an ultra-simple programmer that allows anyone to program the chip using his/her PC. The article 'Free PICProg' elsewhere in this issue comes up with the goods. As usual, the PIC software for his project may be obtained free of charge from our website as file number **030320-11** (see Free Downloads, month of publication) The PIC assembly code file is well commented.

(030320-1)