



Both the BBC Micro and the Commodore 64 use versatile interface adaptor chips to form the user port. Since a user port can be used for both input and output, the port that we are using must first be set to the required mode (in this case, for output). In both micros, this can be done by POKEing the data direction control register directly using BASIC.

Now we must consider how to pulse a user port data line. If a value of hexadecimal 88 (equivalent to decimal 136, or a binary bit pattern of 10001000) were POKEd to the user port, the data pin voltages would be: 5v, 0v, 0v, 0v, 5v, 0v, 0v, 0v, respectively. This pattern would remain until it was deliberately altered. Therefore, a simple pulse can be generated on data line 0 by POKEing hex 00, hex 88, hex 00.

For the pulse to be fast enough, machine code must be used. The algorithm for pulsing a single servo motor is:

- 1) Specify the angle of the lever by storing it in a single byte (called ANGLE) with a value between 0 and 255, using a BASIC program.
- 2) Set data line 0 high (5v), thus starting the pulse.
- 3) Wait one millisecond by looping and decrementing the counter.
- 4) Wait a further period of between 0 and 1 millisecond, again by looping, but this time the starting value of the counter (and therefore the number of loops) is ANGLE.
- 5) Clear data line 0 low (0v) to end the pulse.

If ANGLE = 0, the pulse will last for a duration of one millisecond; if ANGLE = 128 (on the BBC Micro and Commodore 64) it will last 1.5 milliseconds, and the lever will move to a midpoint.

A STREAM OF PULSES

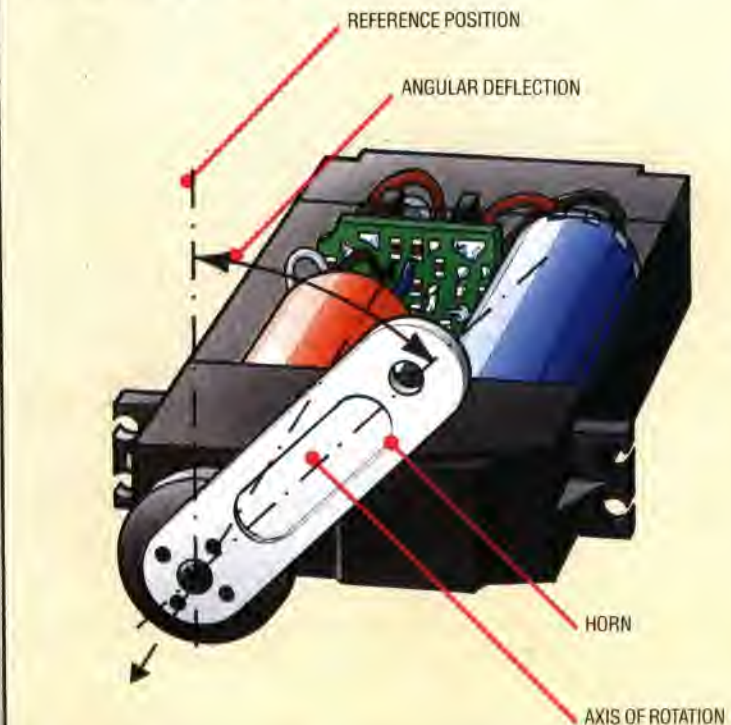
One pulse, however, is not enough to maintain the position of a servo motor. It must receive a stream of pulses, refreshing the motor about every 20 milliseconds. There are two ways to produce a stream of pulses:

- 1) By simply using a wait loop to pause between pulses. But this means that the computer can do nothing else while looping.
- 2) By using 'interrupts', which allow the computer to run another program — usually in BASIC — almost simultaneously. This background program can instruct the motors where to move.

Both the BBC Micro and the Commodore 64 use 6500 series processors, which have two interrupt pins — NMI and IRQ. The second, the interrupt line, will be used for our timing tasks. Whenever a pulse appears on the IRQ line, the processor stops what it is doing and starts to execute the interrupt handling program. When that is completed, it returns (RTI) to where it left off when it was interrupted.

Both the BBC Micro and Commodore 64 use interrupts to run their operating systems. The BBC Micro generates 100 interrupts per second

They Also Serve . .



The deflection of the 'horn' attached to the servo's spindle is determined by the duration of the pulse sent by the controlling computer — the longer the pulse, the greater the deflection. If the pulse is repeated at intervals of about 20 milliseconds, the horn will be held in position by the

motor — this 'position-locking' makes the servomotor ideal for electro-mechanical control applications. Most microcomputers generate interrupts every 10 to 20 milliseconds, so this 'refresh' signal can be sent without reference to a BASIC control program by patching some

machine code into the operating system via the interrupt vector.

A further advantage of the servomotor is that only one data line is needed to send the control and refresh pulses, so only one bit of the computer's data port is committed to each servo.

(one every 10 milliseconds) and the 64 has a rate of 60 per second. On each interrupt the system timers are updated, the keyboard scanned, and so on. Thus, the operating systems of these machines have clocks that generate interrupts, and also have handlers to trap and use them.

In both computers, the system interrupts can be used to run the pulse generating program. The Commodore 64 interrupts must be intercepted by changing the interrupt vector. This vector — a two-byte address held in two consecutive cells — tells the processor the location of the interrupt handling routine. By changing this address to point to the pulser routine, and directing the processor back to the usual system interrupt handler at the end of the pulse, the processor will generate a pulse whenever an interrupt occurs — namely 60 times a second.

We give here the BASIC and Assembly language versions of a program to control a single servo motor on the Commodore 64. In the next instalment of Workshop, we will give listings for control of several motors at once, and the equivalent listings for the BBC Micro.