COMMODORE PULSE GENERATOR

Only three components are needed to make a Commodore C64 into a pulse generator.

JIM BARBARELLO

If your test gear doesn't include a pulse generator it's probably because you just never got around to buying one. Of course, in a pinch you can always use a 555 timer and a few inexpensive components to assemble a quick-and-dirty squarewave or pulse generator. But for about the same cost you can build a simple device that will put your Commodore *64* to work as a stable, accurate source of squarewaves and pulses, and also provide a debounced one-shot trigger source to boot. Actually, the pulse generator consists of the hardware accessory and an accompanying BASIC program.

The software simulates a physical pulse generator. Its screen display combines a digital frequency indicator with a menu for eight functions that are available through the Commodore *C-64*'s normal function keys. No calibration procedure is necessary because the pulse generator uses the computer's 1-MHz crystal-controlled clock for a time base. What you see on the screen is what you get.

Capabilities and limitations

The pulse generator can generate continuous squarewaves in the range of 15 Hz–500,000 Hz, or 1-microsecond width pulses with a repetition rate of 30 pps (*p*ulses *p*er second) to 1-million pps. A *one-shot* function produces a single 1-millisecond pulse on demand. All outputs vary between zero and about 4.3 volts.

The output frequency and waveform is determined entirely by the software. For those of you who might want to experiment with the circuit, we'll take time out to describe how the hardware device uses the Complex Interface Adapter (CIA) IC that drives the computer's user port. With that information and some BASIC programming skill, you can add features such as frequency sweeping, auto sequencing of discrete frequencies, and repetitive trigger pulses having a programmable interval.

The characteristics of the CIA IC require the output frequency to be equal to 500,000/N, where N is a whole number between 1 and 65535. For that reason, the pulse generator's output frequency isn't

continuously adjustable. When you key in a desired frequency the software selects the closest value it can generate. As the frequency increases, the difference between the current and the next frequency value increases. For example, at 100 Hz the next value is 100.02 Hz; at 1000 Hz the next value is 1002 Hz; at 10,000 Hz the next value is 10,204 Hz. Considering that the pulse generator has crystal-controlled accuracy, good resolution in the audio range, and a construction cost of well under $10.00, its performance will adequate for many applications.

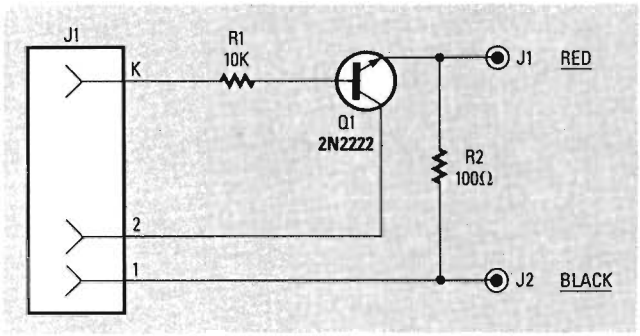

Fig.1—THE USER PORT INTERFACE uses only three components and a connector.

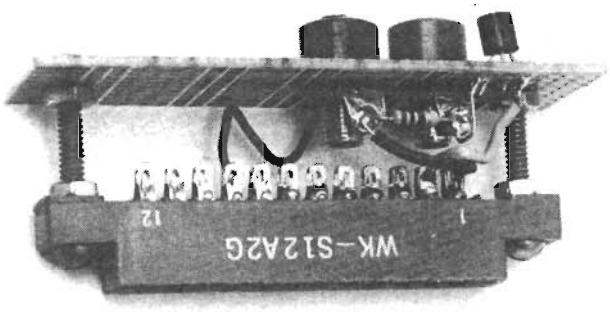

FIG. 2—ALTHOUGH THE LAYOUT ISN'T CRITICAL, try to approximate this layout to insure the interface will fit on the user port.

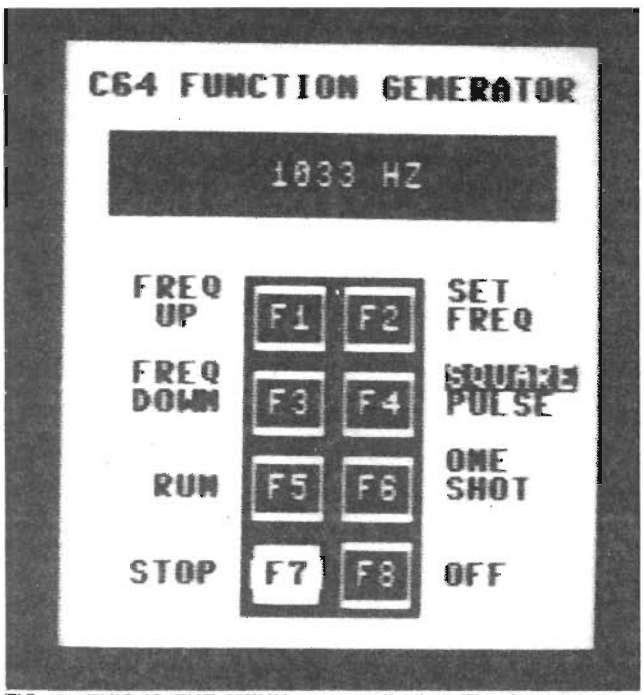

FIG. 3—THIS IS THE MENU screen display. The frequency or pulses-per-second of the output is shown in the rectangle near the top.

The CIA adapter

The Commodore *C-64*'s user port is connected directly to a 6526 CIA, which has two interval timers. The pulse generator uses the one called *Timer A*, which operates just like a standard countdown timer. Before starting, a number representing the count is loaded into the timer. When started, the count begins decreasing by one for each clock cycle. When the count reaches zero, the timer can either stop or reset and begin counting again. Memory locations 56580 and 56581 hold the low and high byte values (respectively) for the count. For example, a count of 1000 would have a high byte value of 3 (the integer part of the product of the count value divided by 256) and a low byte value of 232 (1000 less the high byte value times 256). With a clock rate of 1 MHz, the count can produce either 1000 alternating transitions per second (a squarewave with a frequency of 500 Hz) or 2000 pulses per second.

The value loaded into memory address 56590 controls most aspects of the timer. A value of 2 sets the CIA for pulse output, a value of 3 begins pulse generation, a value of 6 sets the CIA for a squarewave output, a value of 7 begins squarewave generation, a value of 15 produces a single pulse whose width is determined by the value stored in memory locations 56580 and 56581.

Once the timer is in operation, it continues independent of the computer until one of the values in memory locations 56580, 56581, or 56590 are changed. Therefore, all control can be performed directly from the BASIC program by monitoring the contents of those locations.

The hardware interface:

The simple circuit shown in Fig. 1 interfaces the signal from the Commodore's user port to the outside world. Transistor Q1, which functions as a current amplifier, buffers the output from user-port connector J1's pin K (Port B6 of the CIA), an arrangement that allows the signal to drive circuits having current demands that would otherwise distort a direct output from the user port. All output signals appear at banana-type jacks J2 (signal) and J3 (ground).

Operating power is provided by the computer itself from the user port's pins 2 (+ 5 volts) and 1 (ground). The 100-mA maximum rating of the user port allows the circuit to easily drive a 50-ohm load.

Assembly

The circuit is so simple that a printed circuit board assembly isn't necessary. Instead, use a 1" × 3" piece of perforated construction

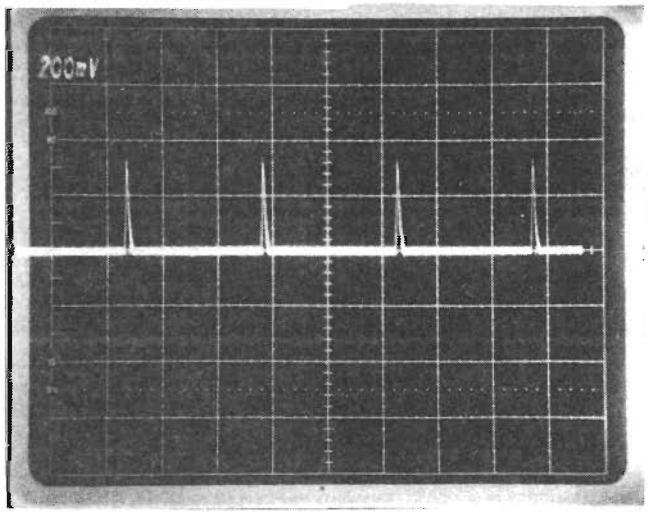

FIG. 4—THE WIDTH OF THE PULSE OUTPUT is so narrow that the signal is changed to a spike by conventional coaxial cable. Use low-capacitance cables and test leads.

```
10 GOSUB 3000:PRINT:F=500:SP=2:P$(1)=" PPS  ":P$(2)=" HZ   "
20 GOSUB 5000
30 COL=10:RO=9:GOSUB5050:PRINTB$" FREQ "W$"┌──┐┌──┐"B$" SET"
35 RO=10:GOSUB 5050:PRINTB$"  UP  "W$"|F1||F2|"B$" FREQ"
40 RO=11:GOSUB5050:PRINTB$"      "W$"└──┘└──┘"
50 RO=12:GOSUB5050:PRINTB$" FREQ "W$"┌──┐┌──┐"B$" "W$"SQUARE"
55 RO=13:GOSUB 5050:PRINTB$" DOWN "W$"|F3||F4|"B$" PULSE"
60 RO=14:GOSUB5050:PRINTB$"      "W$"└──┘└──┘"
70 RO=15:GOSUB5050:PRINTB$"      "W$"┌──┐┌──┐"B$" ONE"
75 RO=16:GOSUB 5050:PRINTB$"  RUN "W$"|F5||F6|"B$" SHOT"
80 RO=17:GOSUB5050:PRINTB$"      "W$"└──┘└──┘"
85 RO=18:GOSUB5050:PRINTB$"      "W$"┌──┐┌──┐"
90 RO=19:GOSUB 5050:PRINTB$" STOP "W$"|F7||F8|"B$" OFF"
95 RO=20:GOSUB5050:PRINTB$"      "W$"└──┘└──┘":GOSUB 500:GOSUB 4000
97 POKE 56590,6
100 CO=12:RO=5:GOSUB 5050:PRINT SP$
110 GET A$:IF A$="" THEN 110
120 G=ASC(A$):IF G<133 OR G>140 THEN 110
130 ON G-132 GOSUB 200,300,400,500,600,700,800,1000
140 GOTO 100
200 CO=17:RO=10:GOSUB 5050:PRINTB$"F1"
210 F=F-1
220 IF F<1 THEN F=1
230 GOSUB 4000:IF PEEK(197)=4 THEN 210
240 CO=17:RO=10:GOSUB 5050:PRINTW$"F1":RETURN
300 CO=17:RO=13:GOSUB 5050:PRINTB$"F2"
310 F=F+1
320 IF F>33334 THEN F=33334
330 GOSUB 4000:IF PEEK(197)=5 THEN 310
340 CO=17:RO=13:GOSUB 5050:PRINTW$"F2":RETURN
400 REM F5
410 CO=16:RO=15:GOSUB 5050:PRINT"▄▄▄":RO=16:GOSUB 5050:PRINTB$"▌F5▐"
420 RO=17:GOSUB5050:PRINT"▀"B$"▄▄"W$"▀"
430 RO=18:GOSUB 5050:PRINTW$"┌──┐":RO=19:GOSUB5050:PRINT"|F7|"
440 RO=20:GOSUB5050:PRINT"└──┘"
450 PT=PEEK(56590):POKE 56590,PT OR 1:RETURN
500 REM F7
510 CO=16:RO=18:GOSUB 5050:PRINT"▄▄▄":RO=19:GOSUB 5050:PRINTB$"▌F7▐"
520 RO=20:GOSUB5050:PRINT"▀"B$"▄▄"W$"▀"
530 RO=15:GOSUB 5050:PRINTW$"┌──┐":RO=16:GOSUB5050:PRINT"|F5|"
540 RO=17:GOSUB5050:PRINT"└──┘"
550 PT=PEEK(56590):CO=21:RO=16:GOSUB 5050
560 POKE 56590,PT AND 14:GOSUB 820:RETURN
600 REM F2
```

PARTS LIST

J1—12/24-pin card edge connector (mating connector for the C64's user port)
J2—Red banana jack
J3—Black banana jack
Q1—2N2222, NPN transistor
R1—10,000 ohms, ¼-watt, 10%
R2—100 ohms, ¼-watt, 10%
Miscellaneous: Perforated construction board, wires, solder, hardware.

NOTE: The 12/24-pin connector (J1) is available for $3.25 each, and the complete program with additional programming information is available on a Commodore-mode disk for $5.00 from B&BTC, RD#1, Box 241H, Tennent Road, Manalapan, NJ 07726. Add $2.00 postage and handling with each order. New Jersey residents must include appropriate sales tax.

board. Any kind of perforated board will do, but the kind having holes spaced at 0.1" intervals and foil pads on one side will make attaching the transistor and resistor easier. In addition to the board material, you will need two 6-32 × 1" round-head machine screws and six 6-32 nuts. Mount the two screws through the mounting holes located on either side of J1. If you're using a standard connector the screws will thread into the holes, making for firm fit. The threaded ends of the screws should be on the same side of the connector as the solder terminals. Secure each screw to J1 with a nut. Drill a hole on both ends of the board about ⅜" up from the bottom edge. Place one nut on each of the screws about 1/4" from the end of the screw.

Temporarily mount the board on the screws and then place one more nut on each of the screws, securing the board about ⅜" away from the ends of J1's terminals. When you are satisfied with the fit, remove the board, cut it to size, install the components on the board, and attach short wires for the connections to J1 pins 1, 2, and K. Reassemble the board to J1 and solder the three wires to the appropriate terminals. The finished unit should resemble the pro-

```
610 CO=20:RO=9:GOSUB 5050:PRINT"▬▬":RO=10:GOSUB 5050:PRINTB$"▮F2▮"
620 RO=11:GOSUB5050:PRINT"▪"B$"▬"W$"▪"
630 CO=0:RO=4:GOSUB 5050:PRINTBB$:PRINTBB$:PRINTBB$
635 CO=0:RO=23:GOSUB 5050:PRINT BL$:GOSUB 5050
640 CO=0:RO=23:GOSUB 5050:PRINT BL$:GOSUB 5050
650 INPUT"ENTER NEW FREQUENCY";F9$:F9=VAL(F9$):IF F9<15 OR F9>.5E6 THEN 640
660 GOSUB 5050:PRINT BL$:F=INT(.5E6/F9):GOSUB 4000
670 CO=20:RO=9:GOSUB 5050:PRINTW$"┌──┐":RO=10:GOSUB5050:PRINT"▮F2▮"
680 RO=11:GOSUB5050:PRINT"└──┘"
690 RETURN
700 REM F4
710 CO=21:RO=13:GOSUB 5050:PRINTB$"F4":CO=25:RO=12:GOSUB 5050
720 IF SP=1 THEN PRINTW$"SQUARE":RO=13:GOSUB5050:PRINTB$"PULSE":SP=2:GOTO 740
730 PRINTB$"SQUARE":RO=13:GOSUB 5050:PRINTW$"PULSE":SP=1
740 CO=21:RO=13:GOSUB 5050:PRINTW$"F4"
750 PT=PEEK(56590):IF SP=2 THEN POKE 56590,PT OR 4
760 IF SP=1 THEN POKE 56590,PT AND 11
770 GOSUB 4000:RETURN
800 REM F6
810 CO=21:RO=16:GOSUB 5050:PRINTB$"F6":GOSUB 5050
815 POKE 56580,232:POKE 56581,3:PT=PEEK(56590)
820 PT=PEEK(56590):POKE 56590,15
830 POKE 56590,PT:PRINT W$"F6":GOSUB 4000:RETURN
1000 REM** END
1010 PRINTCHR$(147):RO=12:CO=6:GOSUB5050:POKE 56590,0:POKE 56579,0
1020 PRINTCHR$(18);" GENERATOR OFF ";CHR$(146);" - PROGRAM ENDED."
1030 PRINT:PRINT:PRINT:END
3000 REM** FORMAT SCREEN=
3010 POKE 53280,6:POKE 53281,6:PRINTCHR$(147)
3020 B$=CHR$(05)+CHR$(18):BL$="          "+B$+"                                    ":PRINTBL$
3030 PRINTTAB(8);CHR$(05);CHR$(18);" C64 FUNCTION GENERATOR "
3040 PRINTBL$:BB$="          "+B$+"  "+CHR$(146)+"                              "+B$+"  "
3050 PRINTBB$:PRINTBB$:PRINTBB$:W$=CHR$(146)
3060 FORI=1TO14:PRINTBL$:NEXT I:PRINT BL$
3070 BL$="                                          ":RETURN
4000 REM ** FORMAT/PRINT OUTPUT
4010 P$=LEFT$(STR$(1E6/(F*SP)),8):P=INT(VAL(P$))
4015 IF P=1E6 THEN P$="1000000":GOTO 4050
4020 IF P<1000 OR P>99999 THEN P$=LEFT$(P$+"     ",7):GOTO 4050
4030 IF P<10000 THEN P$=LEFT$(P$+"     ",5):GOTO 4050
4040 P$=LEFT$(P$+"     ",6)
4050 P$=P$+P$(SP)
4060 CO=16:RO=5:GOSUB 5050
4070 H=INT(F/256):L=F-H*256:POKE 56580,L:POKE 56581,H
4080 PRINT P$:RETURN
5000 REM* CURSOR CONTROL USING PLOT                KERNEL ($FFF0)
5010 DATA 162,0,160,0,24,32,240,255,96,999
5020 A=49300:SC=A
5030 READ B:IF B<>999 THEN POKE A,B:A=A+1:GOTO 5030
5040 RETURN
5050 POKE SC+3,COL:POKE SC+1,ROW:SYS SC
5060 RETURN
```

totype shown in Fig. 2. Be sure to tighten all six screws firmly since you don't want the assembly to flex when you're installing it on the user port. Most 24-pin connectors make a very tight fit to the user port, so make sure all mounting nuts are tight. Finally, install the adapter to the user port.

The software:

The program listing is shown in Listing 1. It is a relatively long program, and if you feel that you're not up to keying in so large a program without making errors you can obtain the program on disk from the source given in the Parts List.

When you run the program, you'll get the screen display shown in Fig. 3. Note that the frequency, which always initializes at 1033 Hz, is displayed in the small dark rectangle at the top of the display. Below the frequency display area are representations of the computer's F1 through F8 function keys, with each key's function clearly labeled. On startup, F7 will be highlighted, indicating that the generator isn't running.

Pressing the F1 key once will increase the ouput frequency one interval. Holding the F1 key down will cause the frequency to continually increase. Similarly, the F3 key causes the frequency to decrease. When the frequency reaches its upper or lower limit, the display will freeze and you will have to reverse the direction of the frequency selection.

Press the F2 key to get to a desired frequency quickly. The F2 screen display will highlight, the frequency display area will clear, and the prompt Enter New Frequency? will appear. Typing any number between 15 and 500000 resets the frequency to the closest

allowable value. Decimal numbers such as 100.56 are allowed, but commas are not (i.e., 500000, not 500,000). If a value outside the working range is entered, it will be ignored and the prompt will be repeated. The display area will then show the selected frequency in Hz or the corresponding pulse rate in pps—and remember, the pulse rate is twice the selected frequency in Hz.

The right function

On startup, the frequency is set to 1033 Hz and the SQUARE function of F4 is automatically initialized. Pressing F4 toggles the output between squarewave (Hz) and pulse (pps). Again, note that the pulse rate is twice the frequency.

Pressing F6 for ONE SHOT generates a single, 1-millisecond pulse. F6 must be released and then pressed again to generate a second pulse. Pressing F8 clears the screen, causes the screen to display the message GENERATOR OFF—PROGRAM ENDED. turns off Timer A, and removes any signal present from the base of Q1 (thus turning it off).

Scope displays

The level and waveform from the pulse generator can be affected by capacitive loading. The most common source of capacitive loading is using a long shielded cable to feed the output to another circuit, or to other test equipment. Normally, high test lead or cable capacitance affects only the higher frequencies. If excessive lead capacitance does exist, the resulting waveform will resemble a triangular wave rather than a squarewave, and the signal level will decrease by as much as 25%. For example, a 6400-Hz squarewave fed through a conventional coaxial-cable test lead had sharp rising and falling edges. However, the signal shown in Fig. 4 also started out as a perfect squarewave, but because its frequency is 500 kHz, the test lead's internal capacitance turned the squarewave into a pulse-shaped wave. To avoid capacitive loading, keep cables short, preferably under two feet, and use a low-capacitance oscilloscope test probe.

The capacitive-loading effect will be even more pronounced on short duration pulses. As shown in Fig. 4, a conventional shielded cable turns an essentially rectangular pulse of 20,000 pps into a thin spike.

Finally, keep in mind that the effective load resistance seen by the adapter should not go below 50 ohms. If you are driving a circuit with an input impedance less than 100 ohms, temporarily disconnect resistor R2 so that it does not parallel the input impedance of the circuit being tested, which would result in a total load of less than 50 ohms. Add an SPST switch if you work with low-impedance circuits often.