EXPERIMENTING WITH RAIL GUNS

DECEMBER 1998

Electronics NOW®

Get accurate signals any time you need them when you build this

Precision Pulse Generator

Check the status of your household power lines, sockets, extension cords and more with an

Electrostatic Voltage Probe

Alternate Energy Sources For Your Home

Will residential fuel cells and microturbines eliminate the need for traditional power distribution?

$4.99 U.S.
$5.50 CAN.

A GERNSBACK PUBLICATION

Give your hands a break with a

Solid-State Soldering-Gun Switch

A MICROCONTROLLER-BASED PRECISION PULSE GENERATOR

Here's a benchtop pulse generator with crystal-controlled timing. What's more, it's built around a PIC microcontroller!

TOM NAPIER

Whether you're developing analog or digital circuits, one of the handiest pieces of test equipment anyone can have is a pulse generator. After all, without some source of signals, most circuits just sit there doing nothing. If you're like most experimenters, you probably find it a little tiring having to hook a timer chip into a circuit in order to test a new idea. As an added drawback, those types of pulse-generator circuits have limits as to their available range and accuracy.

Wouldn't it be nice to have a pulse generator that is inexpensive, accurate, and has a wide range? The Precision Pulse Generator described here is just such a unit. Since it is built around a microcontroller chip, no calibration is needed for pinpoint accuracy when dialing in a particular pulse width or repetition rate. The tradeoff for that ease-of-use feature is that the settings can only be changed in finite steps. However, those steps are small enough so that the unit will be useful on any technician's workbench.

Microcontroller-Based Pulse Generators. In selecting a microcontroller that could do the job of generating accurate and repeatable pulse widths, the PIC 16C55 from Microchip Technologies has several advantages in its favor. At the top of the list is that they are easy to program; the best chip in the world is worthless to the do-it-yourselfer if it can only be programmed with expensive equipment. Another advantage is speed: Most of the various families of Microchip devices can run at speeds up to 20 MHz. However, the most important feature is that each instruction, other than a jump command, has the same execution time. For those who have spent years working with microprocessors such as the Intel 8085 (whose instructions can vary from 1.3 to 5.2 microseconds), it's a pleasant change to find a chip that can generate accurately-timed events just 200 nanoseconds apart.

That 200-nanosecond figure is based on a 20-MHz clock frequency; the instruction rate of a PIC chip is one quarter of that. The main question that then needs to be asked is if a PIC chip could generate an uninterrupted stream of pulses with an arbitrary length and period while still monitoring the front panel controls for any changes. Unfortunately, the short answer is no—unless you want to accept certain restrictions on the pulse period. However, with many months of tinkering with the programming code, it is possible to build a useful bench-top pulse generator with not much more than a Microchip PIC16C55 microcontroller, a crystal oscillator, and four thumb-wheel switches. Specifications for the unit are shown in the sidebar.

The Precision Pulse Generator has both normal and inverted TTL-level outputs. The output's rise and fall times are less than 10 nanoseconds. The timing of the pulse lengths and periods is as good as the accuracy of the microcontroller's clock. A pair of two-digit thumb-wheel switches is used to set the pulse period and the pulse length. A six-position rotary switch selects the range. On the five slower ranges, the pulse period can be set from 1 to 99 units; on the fastest range, the period is settable from 5 to 99 units. The length of a unit, depending on the range selected, varies from 1 microsecond to 100 milliseconds.

The pulse length can be set over the range of 0.1 to 9.9 units. On any range, each step of the pulse-length switch is one tenth of the period switch step. For example, on range 1 the minimum off time is 2.1 microseconds if the period is below 10 microseconds. Above 10 microseconds, it is 4.1 microseconds. At the lower ranges, the minimum off time is 0.1 unit. Those are the limits on the pulse length as discussed above; a front-panel LED warns if any particular switch settings exceed those limits.

Although it might seem that the pulse widths must be less than a tenth of the pulse period, each range is ten times slower than the one above but the switch settings have a 100-to-1 range. The result is that many periods can be set equally easily on either of two ranges depending on the pulse length that you want. For example, if you wanted a period of 50 microseconds, you could set a period of 50 on range 1 (which is one-microsecond units) or a period of 05 on range 2 (which is ten-microsecond units). In the first case, the longest pulse length that you can set is 9.9 microseconds with one-microsecond changes in the period. In the second case, the pulses can be up to 49 microseconds long but the period could only be set in 10-microsecond steps. Because of the limitations, you can't have a pulse with a period of 49 microseconds and a length of 48 microseconds.

A Finer Step. As you can see, it is not too hard to make a PIC-based pulse generator that sets the pulse length in 1-microsecond steps. Naturally, any piece of test equipment will always leave someone asking if the limits can be increased; can the unit be made "better"—and that includes the author!

A desirable minimum-pulse length of 0.1 microseconds is a bit difficult to do if the fastest PIC controller has a 0.2-microsecond instruction cycle. In this case, the problem is solved with the addition of some circuitry. That circuit can be seen in the schematic diagram, which is shown in Fig. 1. The circuit consists of IC5 and IC6.

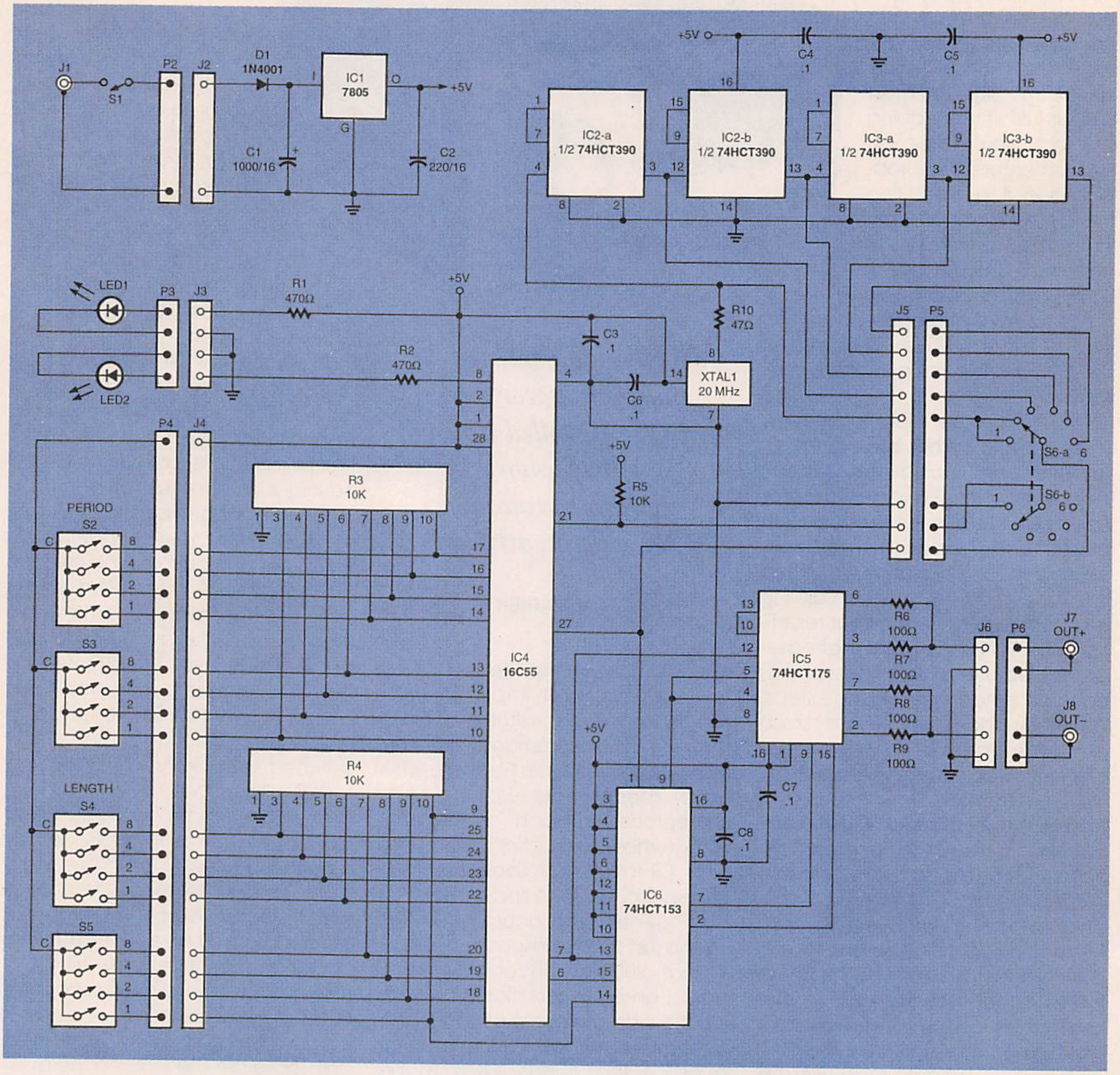

Fig. 1. The Precision Pulse Generator is built around a PIC 16C55 microcontroller. Although it might seem to be impossible, pulses as short as 100 nanoseconds can be generated thanks to a careful balance of hardware and software.

The trick is to add 0.1 microseconds to the pulse length whenever the fastest range is selected and the least-significant length switch is showing an odd number. The amazing part of the solution is that it works even when the switch reads 0.1 and the PIC is not generating a normal output pulse—after all, 0+0.1=0.1!

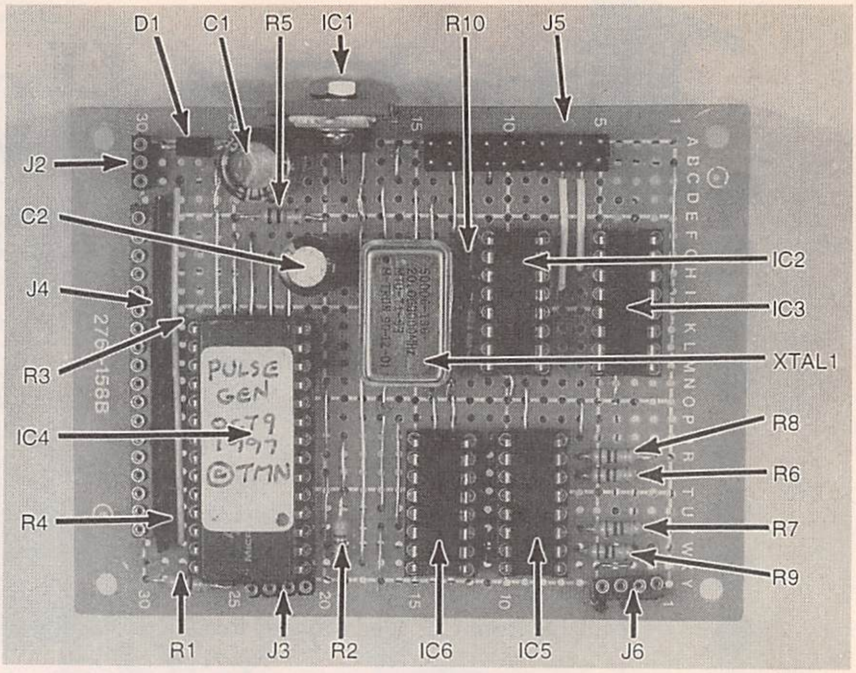

Fig. 2. The Precision Pulse Generator can be easily built on a perfboard. Keep the connections short and direct. Here you can see part of the author's construction technique: the interconnect wires are neatly and carefully laid out in the same direction on this side of the board; the other side has the wires all running at right angles.

Range Switching. The timing for the Precision Pulse Generator is set by XTAL1, a 20-MHz crystal clock. When the two highest ranges are selected, the PIC (IC4) runs from the 20-MHz clock. The clock frequency is sent to IC4 through S6-a. The clock frequency is also divided by IC2 and IC3. Their outputs are also selected by S6-a. At the lowest range setting, the clock frequency is only 2 kHz, making responses to any switch changes a bit sluggish! By using that arrangement, the same software can be used for the different ranges.

The highest range is a bit different. The PIC chip must be told to use a different software program in order to create the extra short pulses with IC5 and IC6. That command is supplied by S6-b. When S6-b grounds pin 21 of IC4, the program knows to use the fine-range technique discussed above.

The clock signals are all sent to the range switch through a length of ribbon cable, and the selected one is returned to IC4. Although that arrangement is not an example of good engineering practice, it does work. Keeping the clock signals on the PC board would have required additional circuitry to switch the various signals, complicating the design. The ribbon cable carries the clock signals on alternating wires, with the unused wires in between grounded. That tends to shield the signals from each other, minimizing cross-talk. A 47-ohm source-termination resistor (R10) is connected to the output of XTAL1 so that the 20-MHz signal is somewhat cleaner when it finally reaches IC4 after its trip to the switch and back.

Hardware, Software, and "Firmware." The 16C55 microcontroller has two eight-bit ports and one four-bit port. The two eight-bit ports are directly connected to S2–S5, with R3 and R4 acting as pull-up resistors so that there is always a valid logic level on IC4's input pins. The software looks at the thumb-wheel switches twice per pulse. The time needed to read and process that information is the reason that the minimum pulse period is five microseconds.

The lowest bit of S5 is not connected to the same input port as the rest of the pulse-length input since it controls the hardware "add-0.1-microsecond" function. Switch S6-b is connected in its place. However, since that bit is needed on the higher ranges, it is hooked up through the controller's four-bit port. It might seem to be an odd arrangement, but it is an example of the balance needed between the hardware and the software in order to get the necessary processing speed.

The software generates the pulses in two stages—the length of "on" time and the length of "off" time. When calculating the length of "off" time needed, the amount of processing time before starting another pulse cycle is taken into consideration. If the amount of "off" time is less than the cycle time, IC4 will light up LED2 to let the operator know that the switch settings are "improper;" that is, the Precision Pulse Generator cannot create the pulse widths and repetition rates that have been set on the front-panel controls.

Pulse Processing. It would be expected that only one pin of IC4 would be needed for the actual output pulse. Since we have additional circuitry that can stretch the output pulse by 0.1 microsecond, two outputs are used. The main pulse output is a positive-going pulse with a 0.2-microsecond resolution. The secondary output is a 0.2-microsecond negative pulse that immediately follows the main pulse with a 0.1-microsecond delay. That pulse drives the lengthening circuit when the units-digit length switch is set to an odd number. Since the secondary pulse is needed only at the fastest range, it is only generated when it is needed.

The lengthening circuit is built around IC6 to generate a pulse that is either the same length as IC4's output pulse or is exactly 0.1 microseconds longer than IC4's output. To do that, a 0.1-microsecond pulse that starts just as the main pulse stops is needed; that is the reason for the secondary output.

Pulse Outputs. In order to provide the most accurate possible timing,

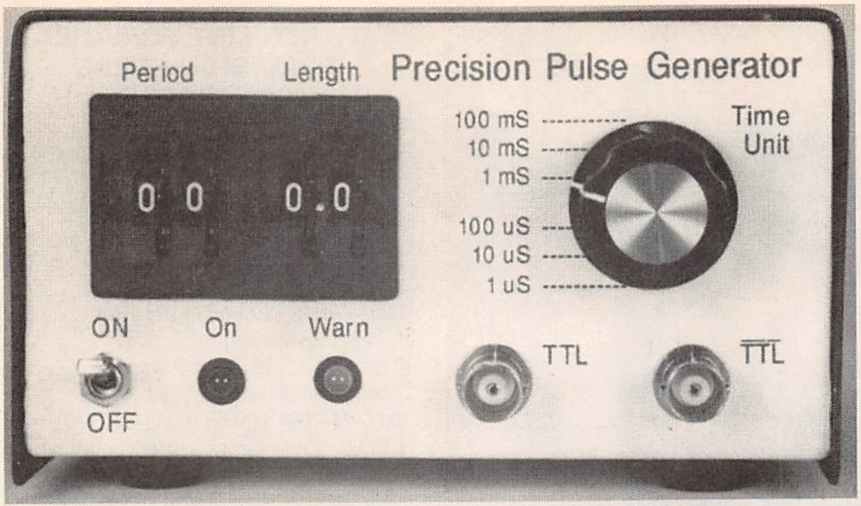

Fig. 3. Like any well-thought-out piece of test equipment, a neat and orderly front panel arrangement is the hallmark of a high-quality unit.

the final pulse from IC5 is resynchronized to the main clock. Two flip-flops in IC5 are tied together in parallel to drive the output load. An added advantage is that the output is available in both inverted and non-inverted form. The HCT-series of chips was selected because their outputs have a fast rise and fall time without excessive overshoot. The AC- and ACT-series are still faster parts, but they tend to have a lot of overshoot on their outputs, and they are somewhat difficult to find in the various "hobbyist-friendly" component distributors.

The outputs are wired to BNC sockets J7 and J8 through R6–R9. The resistors serve three purposes. One is to protect IC5 from momentary short circuits to ground or +5 volts. Another is to share the load current equally between the flip-flops. The third is to provide source termination.

Connecting the output of the Precision Pulse Generator with an unterminated 50-ohm cable will give a reasonably clean TTL-level signal. Using 50-ohm terminators on the cable will speed up the signal edges but leave the pulse voltage level at about two volts—too low to drive a TTL input. The ideal termination would be a 68-ohm resistor tied to ground and a 180-ohm resistor tied to a +5-volt source. That will result in a fast signal capable of meeting the input requirements of TTL.

The normal output-current range of IC5 will not drive a 50-ohm load without risking damage to the IC itself. However, that risk only occurs if the output is shorted to ground or +5 volts. If you are concerned about that situation, you can substitute a 74AC175 for IC5; that family of devices has a higher output-current rating.

Clock Timing. The clock that drives IC5 has to arrive when the flip-flops all have valid inputs. Unfortunately, the designers of the microcontroller didn't specify its clock-to-output delay when it is used with an external clock. After testing the performance of several chips, the author found that the delay from the negative-going clock edge to an output change is about 25 nanoseconds. That is just half the clock period on the fastest two ranges. In that case, the output change occurs at the same time as the positive-going edge of the clock—exactly the worst place for it to be if we want to drive a flip-flop. The cure for that situation is to invert the clock before sending it to the flip-flops. It was at first tempting to use IC4's OSC2 pin as an inverted clock, but that output was designed with a linear output for driving a crystal, which makes it a poor logic signal. Luckily, IC6 has an unused section. By connecting the clock to its enable pin, we not only get an inverted output but we get a little extra delay. That matches the pulse delay through the other half of the chip and makes latching the pulses into the flip-flops just a little bit more reliable.

Power. The Precision Pulse Generator uses only a single 5-volt supply. With a current draw of only 20 mA, it could run on batteries. However, a 9-volt AC or DC wall adapter is connected to J1 as a power source. Even if a DC supply is used, D1 and C1 still provide polarity-reversal protection and additional smoothing. The input voltage is regulated by IC1. The low current demands mean a heatsink is not required for IC1, but a small one can be used just to be on the safe side.

The unit is turned on and off by S1. LED1 indicates whether the Precision Pulse Generator is in the on or off state.

Construction. The Precision Pulse Generator is simple enough to be built on a piece of perfboard using standard construction techniques. Since some portions of the circuit have high-frequency signals with fast rise and fall times, the wiring should be kept as short and as direct as possible. The layout used on the author's prototype is shown in Fig. 2. Note that you cannot see the bypass capacitors C3–C8. Those components are underneath the integrated circuits in order to save space on the board. In fact, the capacitors are built into the IC sockets themselves. Although such sockets save both construction time and space, they are a bit on the expensive side. Using standard sockets and ceramic-disc capacitors will work just as well. After the board is built, do not plug in any chips until after it is tested; set the board aside for now.

Microcontroller IC4 needs to be programmed before being used. A pre-programmed part is available from the source given in the Parts List or you can program a blank part yourself. If you want to program your own part, the compiled code is available for download at the Gernsback FTP site (ftp.gernsback.com/pub/EN/pulsegen.cod).

As with any piece of test gear, the layout of the front panel makes the difference between a unit that is easy to use and one that gets relegated to the top shelf. Details on the author's panel design and layout can be seen in Fig. 3. Most cabinets of the type used here have

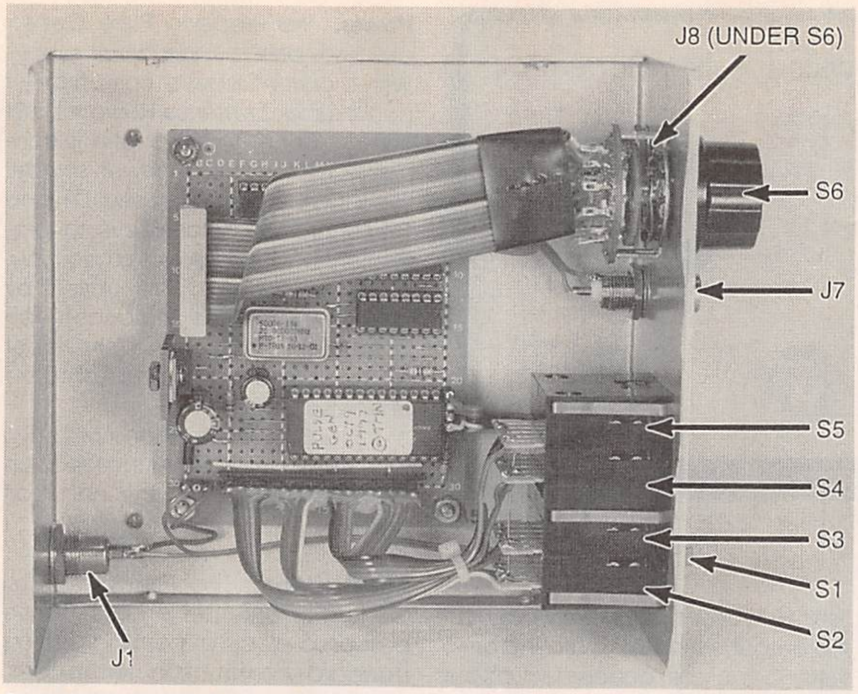

Fig. 4. The inside of the Precision Pulse Generator is neatly laid out thanks to the use of ribbon cable.

aluminum front panels. If yours is like that, be careful when cutting out the rectangular hole needed for the thumb-wheel switches; haste in cutting might result in bending the panel, making the finished product somewhat unsightly.

Once all of the holes in the front panel are cut and drilled, the panel should be labeled. Again, any method that you are comfortable with will work well. The author used a sheet of peel-off self-stick label material and a laser printer. A disadvantage to that method is that the label will become dirty very fast. Spraying a coating of fixative will keep the label looking clean and neat. If you do spray your label, do not use acrylic laquer—it will soak into the paper and destroy the label's glue. Experiment on some scrap before making the final label.

The interior layout of the Precision Pulse Generator is shown in Fig. 4. The connection from J4 to the thumb-wheel switches is most neatly done with a short length of ribbon cable. A socket-plug arrangement can be made from two pieces of strip socket as mentioned in the Parts List. As an alternative, a direct solder connection can be used in place of the socket. However, having a connector makes it possible to take the unit apart should any repairs be needed in the future.

Thumb-wheel switches might look rather old-fashioned in these days of keypads and liquid-crystal displays, but they have the advantage of combining the functions of input, storage and display. The microcontroller doesn't need to store the switch settings; it reads the switches twice per pulse period and uses the result immediately in its calculations. A display is not needed, either; the pulse length and period can be read directly from the switches. The downside is that thumb-wheel switches retail for around $10 a digit. The complete switch assembly, with end pieces and a blank section in the middle will set you back about $50. However, the cost compares reasonably with that of a keypad and LCD display, even if we had the extra processing power to use them.

The switches used here are ten-position Binary-Coded Decimal (BCD) types. It is worth checking the various surplus-dealer catalogues for switches that can be dismantled and reassembled for use in the Precision Pulse Generator.

The thumb-wheel switch assembly consists of seven components stacked together: an end plate, two digit switches, a blank section, two more digits, and a second end plate. The individual pieces have little plastic pins that align them; the stack is held together by two long screws and nuts. If you are a perfectionist, drill a dimple in the face of S5 and fill it with a drop of white paint

PARTS LIST FOR THE PRECISION PULSE GENERATOR

SEMICONDUCTORS

IC1—7805 5-volt regulator, integrated circuit
IC2, IC3—74HCT390 dual decade divider, integrated circuit
IC4—16C55 PIC Microcontroller, integrated circuit
IC5—74HCT175 quad flip-flop, integrated circuit
IC6—74HCT153 dual 4-way multiplexer, integrated circuit
D1—1N4001 silicon diode
LED2—Light-emitting diode, yellow
LED1—Light-emitting diode, green

RESISTORS

(All resistors are ¼-watt, 5% units unless otherwise noted.)
R1, R2—470-ohm
R3, R4—10,000-ohm, 10-pin resistor network, single-inline package (Digi-Key Q9103-ND or similar)
R5—10,000-ohm
R6–R9—100-ohm
R10—47-ohm

CAPACITORS

C1—1000-µF, 16-WVDC, electrolytic
C2—220-µF, 16-WVDC, electrolytic
C3–C8—0.1-µF, ceramic disc

ADDITIONAL PARTS AND MATERIALS

J1—Co-axial power connector
J2–J4, J6— Socket strip
(Digi-Key ED7064-ND or similar)
J5—36-pin 2-row header strip (Digi-Key S2012-36-ND or similar)
J7, J8—BNC connector, panel-mount
P1—not used
P2–P4,P6—Socket strip (Digi-Key ED7064-ND or similar)
P5—20-pin, 2-row plug with ribbon cable (Digi-Key A1AXA-2036M-ND or similar)
S1—Single-pole, single-throw switch
S2–S5—Thumb-wheel switch, binary-coded decimal output (Digi-Key CKN5014-ND or similar)
S6—2-pole, 6-position switch
XTAL1—20-MHz crystal oscillator
IC sockets, thumb-wheel switch end plates, thumb-wheel switch spacer, LED holders, knob, wire, case, 9-volt AC wall adapter, hardware, etc.

Note: A pre-programmed IC4 is available for $12 (includes shipping and handling) from: Tom Napeir, PO Box 3155, Maple Glen, PA 19002-8155. PA residents must add sales tax.

to make a decimal point.

There are two types of mounting arrangements for thumb-wheel switches: those that drop in from the front and snap in place, and those that are bolted in place from behind. The snap-in design is easier to mount as well as covering any imperfections in the panel hole, but the wiring must be threaded through the hole when installing or removing them. The rear-mounted type is more difficult to line up with the front-panel hole, but you don't have to worry about damaging the wires if they should catch on the edges of the panel hole.

Each switch segment on the thumb-wheel switches has five contacts on a little printed circuit board marked "C," "1," "2," "4," and "8." It is easier to solder the ribbon cable to the switch segments before they are assembled into a block. One wire goes to each BCD pin of each switch. A piece of bus wire is threaded through all four common connections after the switch is assembled. That way, only one common wire is needed to connect to the PC board.

In a similar fashion, a length of ribbon cable is used to make the connections from J5 to S6. While you can make your own cable assembly, the Parts List suggests a pre-assembled unit that can be used to save time and effort. Only about six inches of cable will be needed to reach S6; the remainder of the cable can be used to wire up the rest of the unit.

Any two-pole six-way switch can be used for S6. A word of caution on the rotary switches from RadioShack: those units have a 6-mm shaft, making a sloppy fit to their ¼-inch standard knobs unless you either wrap some electrical tape around the shaft or use a metric knob.

Testing. Using an ohmmeter, check for shorts between the power and ground pins of any chip socket. If there are none, it should be safe to connect the 9-volt supply. Check that there is about 12 volts across C1 and very close to five volts across C2. If those tests fail or if IC1 is getting hot, you might have wired D1 backwards.

If all is well at this point, check for 5 volts between pins 8 and 16 on the IC sockets, between pins 7 and 14 of the socket for XTAL1, and pins 4 and 2 on the socket for IC4. Disconnect the power and install XTAL1, IC2, and IC3. With power applied, check pin 4 of IC2 with an oscilloscope; there should be a 20-MHz signal. It should be square but it probably looks more like a sine wave. Pin 3 should have a 2-MHz signal, and pin 13 should have a 200-kHz signal. Moving on to IC3, its pin 3 should show 20 kHz and its pin 13 should show 2 kHz.

When S6 is connected, pin 27 of the socket for IC4 should have a signal that changes between 20 MHz, 20 MHz, 2 MHz, 200 kHz, 20 kHz and 2 kHz as S6 is turned clockwise through its range. Connect S2–S5 and check that the proper voltage levels are reaching the appropriate pins on IC4's socket using either an oscilloscope or a voltmeter.

If everything has tested OK up to this point, you can now plug in IC4. Be careful that none of its pins are bent under during installation. Set S6 to position 2, set the period to 10, and the length to 2.0. Pin 6 of IC4 should show a pulse that is positive for 20 microseconds and zero for 80 microseconds. When you change the settings on S2–S5, the pulse length and period should change correspondingly. If you set a period that is less than or equal to the length, LED2 should light. That test can be repeated for the other low ranges. It should show the correct pulse lengths and periods on all of them.

Switch the unit to the highest range. Set up a period of 10 microseconds and a length of 2.0 microseconds. Pin 6 of IC4 should show a 2.0-microsecond positive pulse and pin 7 should show a 0.2-microsecond negative pulse immediately following it. When you change the lower digit of the length switch (S5), the pulse length should only step on every second position. The negative pulse on pin 7 should always be present even when the length is set to 0.0. Check pin 9 of the IC6; the output pulse should be negative-going with a step that changes with every digit change.

If you connect an oscilloscope to either J7 or J8, you should see that the output pulses match the front-panel settings. Check also that all of the illegal settings, light LED2. If everything checks out, the Precision Pulse Generator is ready to use. Happy pulsing. Ω

Precision Pulse Generator Specifications

Pulse output level:	TTL and complementary TTL unterminated 2 volts with 50-ohm termination
Rise/fall times:	Less than 10 nanoseconds
Timing precision:	100 ppm ±2 nanoseconds

	Pulse period	Pulse length
Range 1	5–99 microseconds	(See below)
Range 2	10–990 microseconds	1–99 microseconds
Range 3	0.1–9.9 milliseconds	10–990 microseconds
Range 4	1–99 milliseconds	0.1–9.9 milliseconds
Range 5	10–990 milliseconds	1–99 milliseconds
Range 6	0.1–9.9 seconds	10–990 milliseconds

Range 1: Pulse period	Pulse length
5 microseconds	0.1–2.9 microseconds
6 microseconds	0.1–3.9 microseconds
7 microseconds	0.1–4.9 microseconds
8 microseconds	0.1–5.9 microseconds
9 microseconds	0.1–6.9 microseconds
10 microseconds	0.1–5.9 microseconds
11 microseconds	0.1–6.9 microseconds
12 microseconds	0.1–7.9 microseconds
13 microseconds	0.1–8.9 microseconds
14 microseconds and above	0.1–9.9 microseconds

A square wave output can be set for all periods.