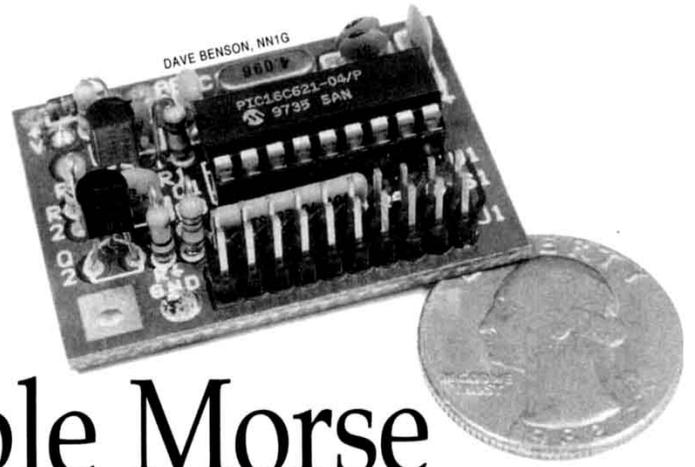


By Dave Benson, NN1G

FREQ-Mite— A Programmable Morse Code Frequency Readout



This simple, one-evening project delivers a rig's frequency with dots and dashes!

When I incorporated a Morse-based frequency readout in a recent design,¹ I received a number of inquiries about its use in other applications. The PIC microcontroller family lends itself beautifully to a wide variety of Amateur Radio projects, and the temptation to modify that readout design for general-purpose use proved irresistible!

General Description

For experimenters and even occasional "home-brewers," a simple frequency counter often comes in mighty handy. At a mere 0.4×1.75×1.25 inches (HWD), the FREQ-Mite is small enough to get lost in your shirt pocket! For many of the QRP crowd, portable and backpacking rigs attract a lot of interest, and the compact form factor of this counter is aimed at ease of incorporating these gadgets into field-ready gear.

Morse code is unbeatable for hardware simplicity. To get this compact, this counter's output is delivered as an 800 Hz Morse code audio signal. The audio signal can be injected into the AF portion of a receiver, transceiver or, for stand-alone applications, connected to a piezo annunciator or speaker.

In the title photo you'll see a dual row of connector pins. By properly interconnecting them, these pins define an operating offset for the counter in the event it's embedded in, say, a superhet transceiver. In a superhet configuration, a local oscillator (LO) and mixer convert the incoming signal to an intermediate frequency (IF). In

the transmitting mode, the same LO signal is mixed with an IF signal to form the transmitter's output frequency.

A wide variety of IF choices exists in kit-based and home-brewed gear. As a result, a general-purpose counter such as this needs a means of providing a frequency offset for a number of the possible IF values. In earlier designs, this process was handled by using analog trimmer potentiometers,² preprogrammed offset look-up (for the most popular frequencies used in kits) and even by keyer-paddle input. In this counter, the rows of jumpers (10 in all) serve as a compact way to enter any offset value (0-999) for superhet transceiver use. Because you already know which amateur band you're operating in (I hope!), there's little to be gained by announcing the megahertz digit(s). In the transceiver mode, then, the counter simply outputs the frequency in

hundreds, tens and units of kilohertz. For example, with an operating frequency of 7.112 MHz, you'd hear **112** in Morse when requesting a readout.

These inputs also serve to select the general-purpose mode: Jumpering the left-most seven pins on the connector strip corresponds to an invalid offset and the readout is in full four or five-digit fashion.

The counter is programmed to count any input frequency up to 32.767 MHz. Sensitivity is good. Figure 1 shows the minimum input signal required as a function of frequency. Accuracy is quite acceptable as a general-purpose tool—it's within 2 kHz of measurements made with other counters over this frequency range. At the other extreme, I've tested the circuit with inputs of up to about 10 V P-P input without incident. Somewhat higher drive levels (within reason) shouldn't harm this circuit. The maximum usable input-signal level begins to fall off around 25 MHz as parasitic effects become more prominent, but reducing the RF-input signal to a more modest level takes care of this situation.

Circuit Description

As a glance at the schematic (Figure 2) reveals, there's not a lot of complex hardware in this design! The Microchip 16C621 PIC^{3,4} (U1) is a single-IC, 8-bit microcontroller. U1 requires 5 V dc, which is ensured by the combination of R5 and Zener diode D1. Q1 is an emitter follower. It presents a high-impedance load to the given RF source. This is important if the counter is connected directly to a VFO as you don't want the counter loading (pulling) the oscillator frequency. Q2 is configured as a high-gain amplifier and converts

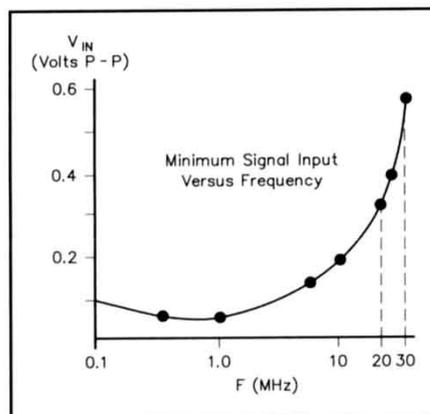


Figure 1—Input sensitivity versus frequency.

¹Notes appear on page 36.

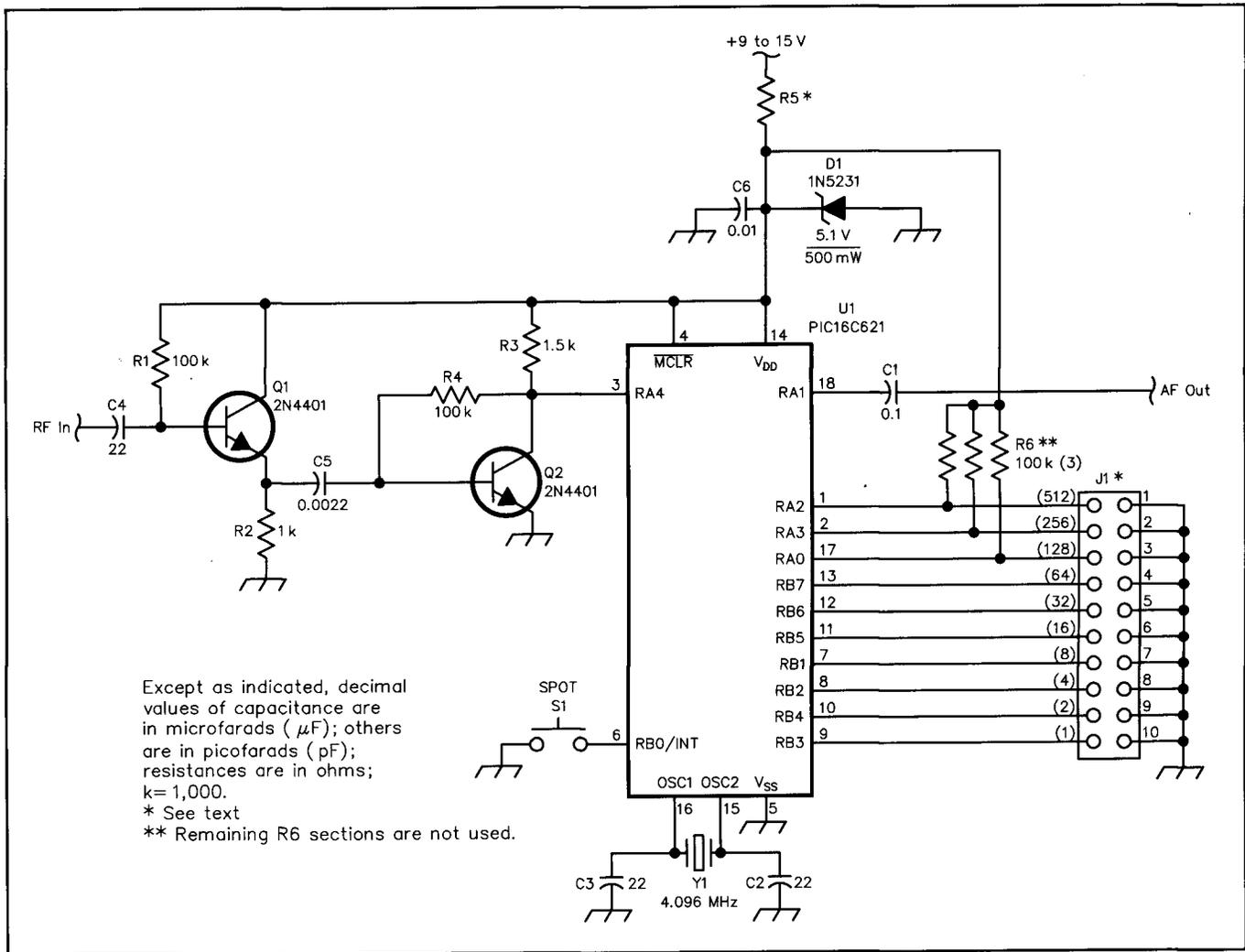


Figure 2—Schematic of the FREQ-Mite. Unless otherwise specified, resistors are 1/4 W, 5% tolerance carbon-composition or film units. Equivalent parts can be substituted.

- C1—0.1 μF monolithic cap
- C2-C4—22 pF NP0, 5%-tolerance ceramic disc
- C5—0.0022 μF disc ceramic
- C6—0.01 μF disc ceramic
- D1—5.1 V, 500 mW Zener diode (1N5231)
- Q1, Q2—2N4401 NPN
- R1, R4—100 k Ω

- R2—1 k Ω
- R3—1.5 k Ω
- R5—330 Ω or 470 Ω (see text)
- R6—100 k Ω resistor network, three sections used (6-lead/5-resistor SIP such as Digi-Key 770-61-R100K [CTS] and Q5104 [Panasonic]); Digi-Key Corp, 701 Brooks Ave S, Thief River Falls, MN

- 56701-0677; tel 800-344-4539, 218-681-6674, fax 218-681-3380; <http://www.digikey.com>.
- U1—Preprogrammed PIC 16C621 (see Note 3)
- Y1—4.096 MHz, HC-18/U holder (20 pF)
- Misc: 18-pin IC socket, hardware.

the RF signal to a 5 V P-P wave compatible with the PIC's I/O requirements. Audio output is taken from pin 18 of the PIC and consists of an 800 Hz square-wave tone sequence. This output is activated only on power-up and on closing S1 (SPOT) for a frequency reading. Otherwise, the output is turned off to minimize "thump" when this project is tied to an audio circuit.

When power is applied, U1's code performs a number of initialization chores. First, it examines each of the J1 inputs (jumper positions) to form a value used later to calculate IF offsets. (How these jumpers are set is covered later under **Construction, Setup and Use**.) U1 then asks for input: readout speed (slow or fast) and for normal or inverted readout, if for

transceiver use. Following these steps, U1 goes to sleep.

When you press the SPOT switch, U1 wakes up and counts the number of LO cycles for a period of one millisecond, in effect, the LO frequency in kilohertz. (Actually, a divide-by-128 prescaler function is used and the main timebase counts for 128 milliseconds.) If the offset jumpers have established that the chip is in a transceiver (three-digit) mode, it adjusts the count result for the programmed IF. If the IF offset is jumper-programmed to an invalid offset (greater than 1008 [decimal]), the counter output switches to a full four-to-five-digit readout and no offset is applied. A math routine within U1 converts the count result from its native hex format

to binary-coded decimal (BCD). These data are the values of the 10 MHz, 1 MHz, 100 kHz, 10 kHz and 1 kHz digits. These digits are then converted to Morse strings using a look-up table. The strings are fed to a routine that toggles U1's I/O port RA1 (pin 18) at an 800 Hz rate; the audio is then sent to the associated equipment's AF amplifier. Output audio is programmed at 13 WPM (slow) and 26 WPM (fast); speed selection is discussed later.

Construction, Setup and Use

PC boards, programmed PICs and kits of parts are available (see Note 3). When building your FREQ-Mite, note that the value of R5 should be 330 Ω for use with a 9 V power supply and 470 Ω with a 12 to

15 V supply. Make sure that the power supply is disconnected before you install U1. *This part is static-sensitive. Ground yourself while handling and installing this device.* One easy way to do this is to use a conductive wrist strap.

In receivers and transceivers, the point at which the *FREQ-Mite's* output is injected into the radio's audio chain is typically just prior to the radio's audio-output stage. A 100 kΩ to 10 MΩ resistor connected in series with C1 in the output line to the host audio stage reduces the signal to acceptable levels. You'll likely have to experiment with the resistance value as the required audio level is rig-dependent.

Programming

The *FREQ-Mite* can be programmed for use with most transceivers. Transceivers generally use a superhet configuration in which the LO is shifted by the IF to yield the operating frequency. The *FREQ-Mite* can be set to any three-digit offset (0 to 999) by means of jumper block J1. The row of header locations is binary-weighted. The right-most position (position 10) has a code of 1, the one adjacent to that is 2, then 4, etc. The left-most position has a code of 512.

Transceiver (Three-Digit) Use

The offset code should be set equal to the three-digit value of the IF in kilohertz. *You may need to vary this value slightly (±1 kHz) to compensate for offsets within the rig.*

Example: If the IF is 8.192 MHz, 8.192 = 192. The binary-weighted code for 192 = 128+64, the jumper coding (from left to right on the board) is 0011000000. You simply add jumpers at the locations denoted by 1 and leave the 0 locations open.

Programming Example for Inverted Operation

The White Mountain-75 SSB rig uses a 9830 kHz IF. The offset coding is set to 830, and the pushbutton should be pressed after the *I?* prompt. For instance, for an LO of 6005 kHz, the *FREQ-Mite* calculates (9830 - 6005) and outputs 825 to indicate an operating frequency of 3.825 MHz.

Naturally, there are exceptions: In the case of a rig with a 455 kHz IF and an LO above the operating frequency, the programming must be reversed. For example, if the IF is 455 kHz and the LO is 4055 kHz, the operating frequency is 3600 kHz. The correct offset value is (1000 - 455) or 545.

Sanity Check

After you've checked your *FREQ-Mite* for correct assembly, install jumpers at J1 positions 3 and 4 (the 128 and 64 binary-weighted positions). Connect the audio output of the *FREQ-Mite* as detailed earlier. With no RF source connected to the *FREQ-Mite*, apply power to the circuit. The *FREQ-Mite* will output *S?* in Morse. Wait until U1 sends an *AR* character to let you

know the initialization is complete. Briefly press S1 and the Morse readout you hear should equal 192. If you want to check out the various binary codes, you'll need to cycle power to the board each time you change jumpers; the offset is read only during power-up.

General-Purpose Use

When you're home-brewing an oscillator or checking a transmitter's output, it's useful to know the operating frequency complete with the megahertz digit(s). To enter this mode, add shorting jumpers to the left-most seven shorting locations of J1 (positions 1 through 7) and apply power. This results in an offset code of 1008 (decimal), and the chip recognizes this as an invalid offset, suppresses the offset calculations, and outputs four or five digits. Here are some examples: For an input frequency of 455 kHz, U1 sends 0455; for an input frequency of 7.110 MHz, U1 sends 7110; for an input frequency of 21.106 MHz, U1 sends 21106.

Initialization

Readout Speed

On power-up, *FREQ-Mite* sends an *S?*. If you press S1 within about two seconds, U1 switches to a higher speed (26 WPM) readout. It acknowledges this entry with an *R*. If you do nothing, U1 maintains its default (13 WPM) readout speed.

Normal/Inverted Readout

If—and only if—you've programmed a legitimate offset for transceiver (three-digit) mode, *FREQ-Mite* inquires if you want an inverted IF by sending *I?*. If you press S1 within about two seconds, the chip switches to calculating an inverted frequency. (This is necessary if the IF is above the LO because the operating frequency is the difference between these two frequencies rather than the sum.) In this case, U1 acknowledges this entry with an *R*. If you don't press the pushbutton, the chip maintains its default (summing) operation. In this case, the chip sends an *AR* prosign after several seconds to let you know it's completed initialization.

Acknowledgments

Wayne Burdick, N6KR, deserves much credit for the concept of annunciating operating frequency in Morse code. Thanks also to Tony Fishpool, G4WIF, and Jim Hossack, W7LS, for their constructive comments and suggestions during development.

Notes

¹Dave Benson, NN1G, "A Single-Board QRP SSB Transceiver for 20 or 75 Meters," *QST*, Apr 1997, pp 29-33.

²Neil Hecht, "A PIC-Based Digital Frequency Display," *QST*, May 1997, pp 36-38.

³This project requires a programmed 16C621 PIC. Programmed PICs and assembly instructions are available for \$12 postpaid; a complete kit of parts, including a double-sided PC board, all on-board parts, mounting hard-

ware and instructions is available for \$20 postpaid from Small Wonder Labs, 80 E Robbins Ave, Newington, CT 06111; bensondj@aol.com. Only personal checks or money orders are accepted. Source code is not available.

⁴See John A. Hansen, W2FS, "Using PIC Microcontrollers in Amateur Radio Projects," *QST*, Oct 1998, pp 34-40.

Dave Benson, NN1G, is well-known to the QRP community. He's been publishing articles for specialty journals since the early '90s, and his work has appeared in QST and The ARRL Handbook. Dave owns Small Wonder Labs, a supplier of QRP-oriented Amateur Radio kits. When not dreaming up new projects, his other interests include gardening, camping and playing guitar. Dave also reports—with considerable relief—completing an addition to his home.

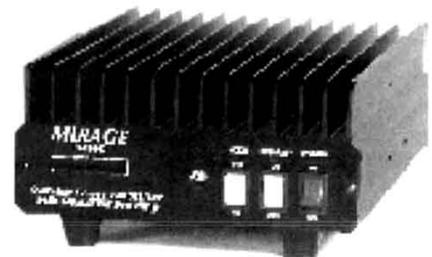
You can contact Dave at 80 E Robbins Ave, Newington, CT 06111; bensondj@aol.com. 

New Products

DUAL-RANGE 2-METER AMP FROM MIRAGE

◇ Mirage's new B-320-G *BruteFORCE* amplifier can boost your hand-held or mobile rig to a beefy 200 W output on FM, CW and SSB. Two power input ranges, 0.25 to 10 W, and 10 to 50 W, assure almost universal compatibility. Features include a 15-dB GaAsFET preamp; SWR and reverse-polarity protection; RF-sensed T/R switching; and an LED power-output bar graph.

The amp is housed in a 7.75x2.75x9.75-inch enclosure with a wrap-around heat sink and a quiet cooling fan. The B-320-G ships with a mobile mounting bracket, a patch cable and rubber feet (for indoor use).



Price: \$449.95. The amplifier is backed by Mirage's one-year warranty. For more information, see your local Amateur Radio products dealer or contact Mirage Communications, 300 Industrial Park Rd, Starkville, MS 39759; tel 800-647-1800, fax 601-323-6551. mirage@mfjenterprises.com; <http://www.mirageamp.com>. 