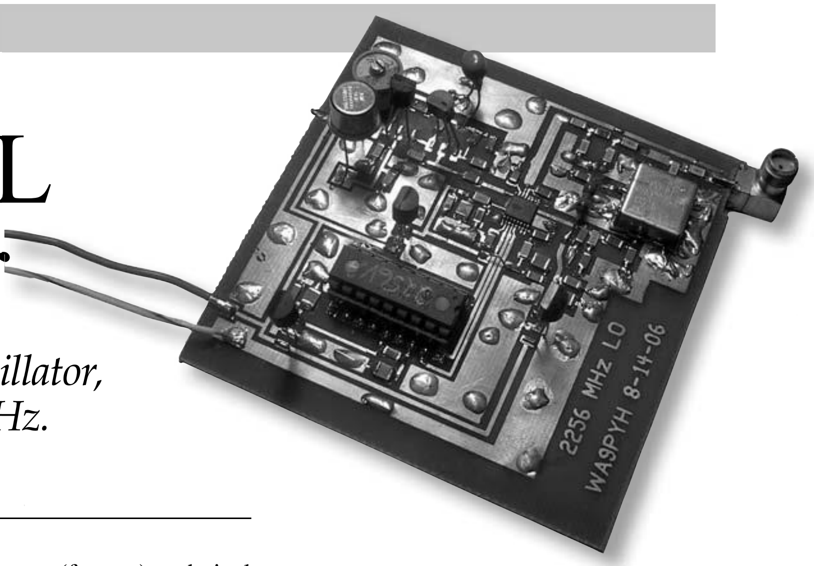
Jim Kocsis, WA9PYH

53180 Flicker Ln, South Bend, IN 46637, wa9pyh@arrl.net

A 2256 MHz PLL Local Oscillator

Using a 10 MHz crystal reference oscillator, this PLL tunes a VCO to 2256 MHz.



This article describes the design of a PLL based local oscillator that operates at 2256 MHz. A PIC16F84 PIC (programmable interface controller) is used to load data into a National Semiconductor (NS) LMX2326 PLL (phase locked loop) IC. Using a 10 MHz crystal reference oscillator, the PLL tunes a VCO to 2256 MHz.

I'll describe the operation of the PLL IC, the software that the PIC loads into the PLL, and most importantly, the operation of software on the National Semiconductor Web site that I used to design the very critical loop filter. I will also discuss building the LO, showing the construction sequence and some areas that can cause problems. A spectrum analyzer display of this LO and an HP generator (as a standard) shows the relative purity of this LO.

The cost of all parts if purchased new is approximately $100. Some savings can be realized if you have a well-stocked junkbox, or if you obtain some of the parts at hamfests. Construction time is approximately 10 hours. No special test equipment is needed, but if you have access to a spectrum analyzer, it can be used to check for proper operation, signal purity and the level of the output. A simple frequency counter is used to adjust a crystal oscillator at 10 MHz and a standard DMM is needed to check the voltage regulators and VCO tuning voltage. Unlike other high frequency oscillators that use multipliers, this circuit needs no tuning or peaking of tuned circuits — there are none!

This LO can be combined with a mixer, LNA and filter to make a down-converter for the 2.4 GHz amateur band with 2 meter output. I built a down-converter using this LO and its sensitivity is equal to that of a commercially available unit with a 1 dB noise figure. When developing the down-converter, the mixer and LNA were easy to design compared to the LO. A ham friend warned me that the LO would be, by far, the most difficult part of a down-converter. Boy, was he ever right!

A Bit of History

I enjoy trying new (for me) technical projects. I wrote an article in the Mar/Apr 2000 issue of *QEX* describing my design of a down-converter for 1691 MHz weather satellite reception.[1] I wanted to apply what I had learned from that design to a 2.4 GHz down-converter. I thought of modifying that design for operation at 2256 MHz, but while the LO I described in that article worked, I was not happy with the noisy output. I had not understood loop filter design and some layout issues as well as I do now. It was basically cut-and-try for the loop filter resistor and capacitor values, until it was "good enough" to get pictures from the weather satellites. The PLL IC used in that design, a Motorola MC12179D, came with a Microsoft *Excel* file for loop filter design. It was difficult to understand and use. That was 1997. Moving up to 2005, I discovered that National Semiconductor provides a very complete and easy-to-use Web site — called "WebBench/*EasyPLL*." It is a true masterpiece and it's free! (It is not software that can be downloaded — you must run it on their Web site.) Considering the function it is designed to perform (loop filter design), it is about as simple as you will find. I knew from my reading and experience that the design of a PLL loop filter would either make or break the performance of my new down-converter. If the loop filter design is not proper, you won't hear weak signals and those that you do hear will have noise on them. I found that National Semiconductor makes a PLL IC that fits my needs exactly — the LMX2326. Besides, Motorola discontinued production of the MC12179D PLL a few years ago.

About this same time I saw an article describing a down-converter that used the LMX2326, and operated around 1600 MHz.[2] It uses a PIC to program the PLL. This was proof that I was on the right track when I chose the LMX2326. I didn't have any experience with a PIC, so I learned the basics of programming, changed the code in that article and programmed my own PIC.

Before using *EasyPLL*, I had tried manually computing the loop filter values using methods described in magazine articles and application notes. With each manual method, however, I always hit a "brick wall." Progress stopped when they didn't provide all the information needed. They assumed I knew what default values to use, what units to use, and so on, so I was never able to finish the design using the manual methods. I quickly switched to *EasyPLL* and began seeing real progress in filter design. If you've entered valid numbers, *EasyPLL* shows a "Congratulations on designing your loop filter!" message. That message is *very* encouraging when you're just getting started!

Please note that the LMX2326 PLL IC is *very* small. See Figure 1. The IC body measures 0.2 × 0.17 inch with eight pins on each of two sides. The pins are 0.016 inch wide and spaced 0.026 inch center-to-center. If you're not used to working with parts this small you may find it a challenge to build this project. You'll need steady hands and good vision (probably with the help of a magnifier/glasses) to solder the PLL IC onto the circuit board. You might also want to ask for help from someone with more experience working with surface mount ICs.

Figure 1 — The LMX2326 PLL IC is *very* small, as this photo shows.

[1]Notes appear on page 25.

Theory

A block diagram of the circuit is shown in Figure 2, and Figure 3 shows the complete schematic diagram. The reference oscillator (exactly 10 MHz) is divided by 5, and the VCO output (approximately 2256 MHz) is divided by 1128 inside the PLL. Both divided signals are combined in the phase comparator at 2 MHz. The charge pump in the PLL produces pulses that are filtered by the loop filter to drive the VCO to exactly 2256 MHz. For more information on PLLs please review the PLL section of recent versions of *The ARRL Handbook*, the Mini-Circuits Web site and a downloadable book from National Semiconductor called *PLL Performance, Simulation and Design*, also known as "Dean's Book."[3, 4, 5] They are all excellent, and are required reading if you want to understand PLL operation fully. Other excellent articles are listed in Notes 6 through 11.

When using *EasyPLL*, there are HELP pages available for almost every parameter. I suggest that you print out all the HELP pages in *EasyPLL*, and put them into a 3-hole punched 8½ × 11 inch notebook for easy reference. I found this to be much easier than opening each file as needed and switching back and forth between them. I added my own notes to each page to help me retain what each page discussed. Also, if a paragraph didn't apply to this project, I crossed it out.

When designing a loop filter for fast switching between frequencies (such as in cell phones), there are trade-offs between signal purity and switching time. Since switching time is not a concern here (the LO operates at a fixed frequency) the design can be optimized for purity by setting Optimize to "Max High Order Cap" in *EasyPLL*. In the example in the "Setup and Using *EasyPLL*" sidebar, the value for Max Lock Time is set to 300,000 microseconds.

When designing a receiver or down-converter, you must pay very close attention to noise in the LO, since any noise there is combined with the desired signal. Sources of noise are: poor loop design (wrong values), long Vtune and power supply printed circuit traces that act as antennas, VCO not loaded with 50 Ω, poor power supply bypassing/filtering and poor grounding of the VCO case and ground leads.

I found a wide-range VCO to be trouble. Consider that when using a VCO with a range of 150 MHz / V, 100 µV of noise can modulate the carrier 15 kHz! My first design used such a VCO, and I was unable to attain a clean signal. I changed to a narrow range VCO (15 MHz / V) and was then finally able to attain a clean signal. I read in several of the references that wide-range VCOs have more noise on their output as compared to narrow range units.

In an early version of my LO, which used a wide-range VCO, I also had long circuit traces between the loop filter and the Vtune input to the VCO (a poor design). I'm sure this contributed to my difficulty with attaining a clean output.

Impedance matching of the VCO output is very important. On the Z-Communications Web site I found that for lowest noise the VCO should "look into" as near a pure 50 Ω load as possible.[12] The three 18 Ω resistors at the VCO RF output and the 50 Ω loads they in turn "look into," result in a 50 Ω load being presented to the VCO. See the Z-Communications Application Note, AN-102 "Proper Output Loading of VCOs." (See Note 12.)

Signal levels at each pin are important. If the signal level is too low at a given point, the circuit won't work. If it's too high it may not work or you may damage an IC. The circuit uses an attenuator to adjust the level of the 2256 MHz feedback to well within the limit shown on the data sheet. The attenuator network (R9-R11) has a 50 Ω input and output impedance. Signal levels are shown in the block diagram in Figure 2.

The reference oscillator (10 MHz in this design) must be very clean, with very low phase noise. Any noise in the reference oscillator will show up in the LO output at a level 20 Log N dB higher. (N is the divide-by ratio in the PLL — in this case 1128.) Any drift in the reference is also multiplied by N, but since most crystal oscillators are very stable this isn't a big concern. The reference oscillator in this LO is a crystal oscillator from a *QST* Technical Correspondence Letter by John Clark, K2AOP.[13] I was looking for a good, clean circuit when I saw this article. What caught my attention was that the phase noise was so low that the ARRL laboratory couldn't measure it — that was good enough for me so I incorporated it into my LO!

A VCO running "open loop" has a very broad noise spectrum that is too high to allow it to be used as-is for a local oscillator. VCOs must be locked to a very clean reference to

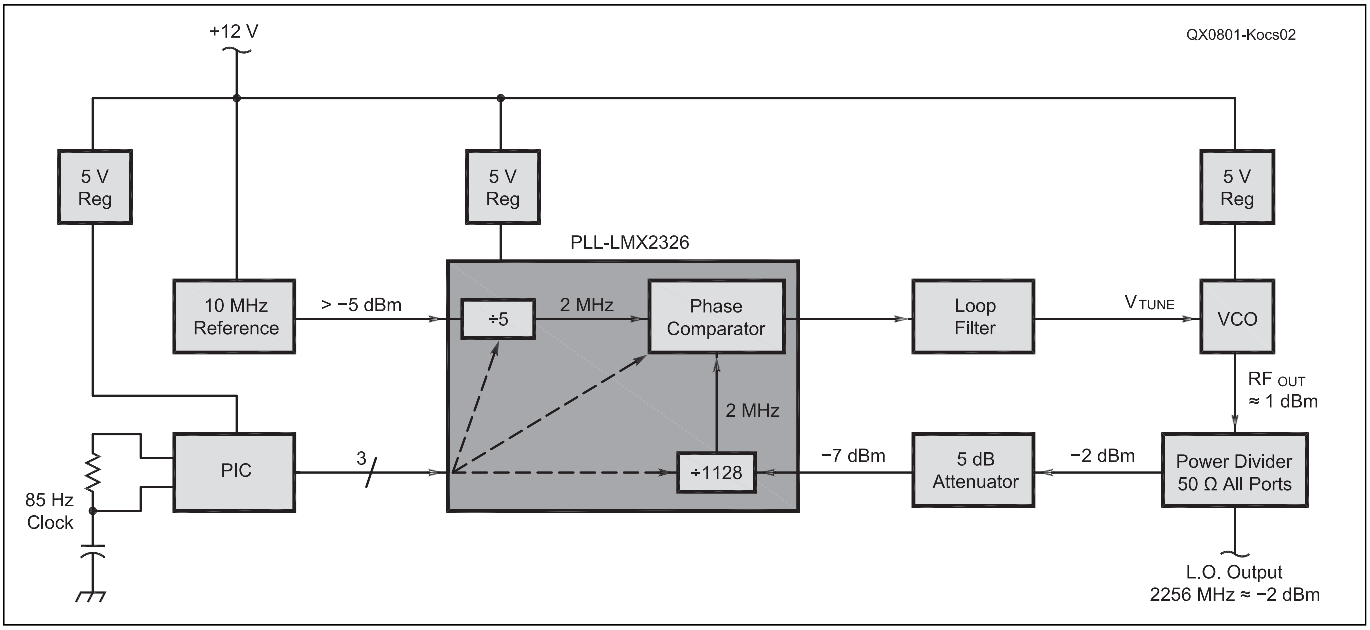


Figure 2 — This block diagram shows the operation of the local oscillator circuit.

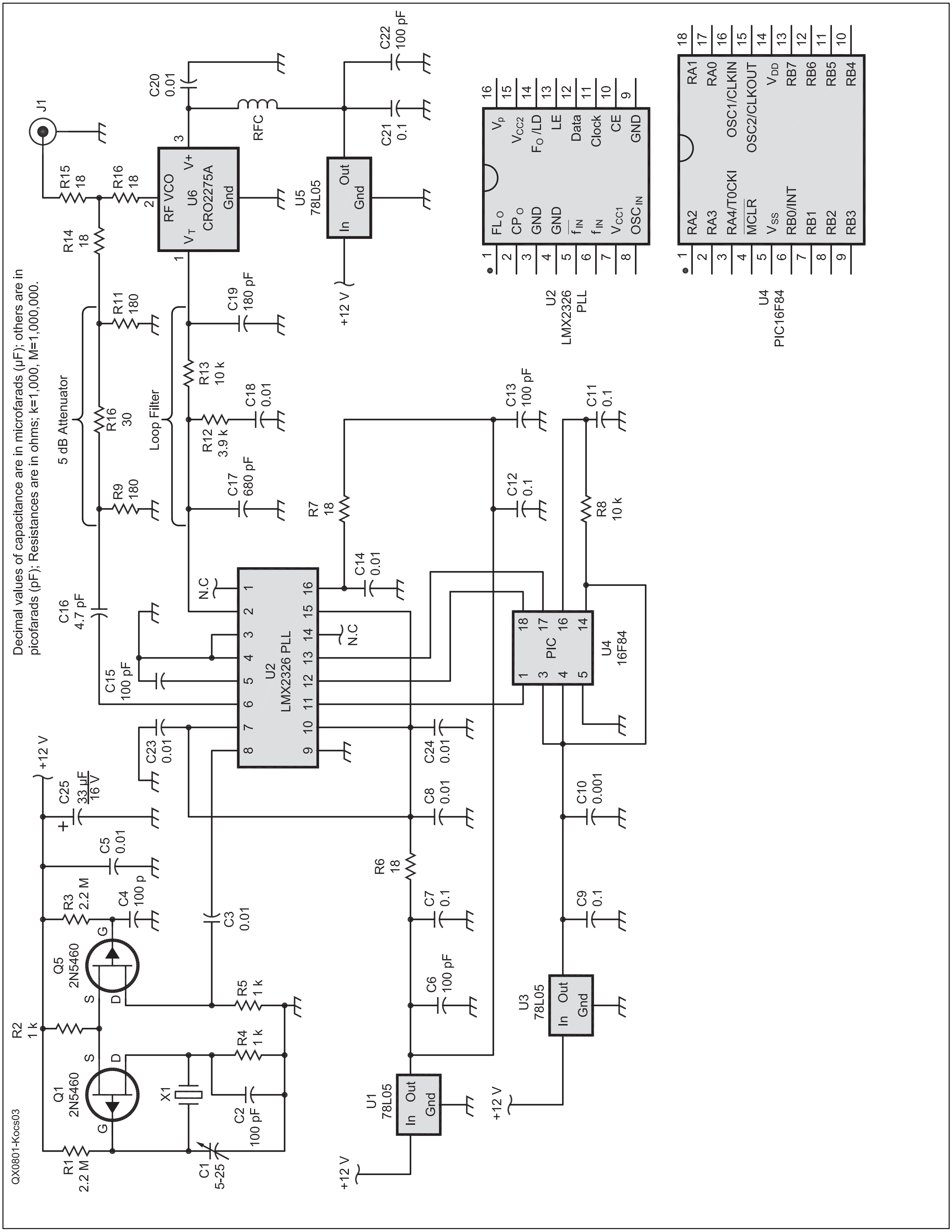


Figure 3 — Here is the schematic diagram of the local oscillator circuit.

produce a clean output. Also, they drift a lot unless "locked" to a stable source. The loop filter and PLL IC actually reduce the noise in the sidebands of the VCO output — up to the bandwidth of the loop filter. Beyond the loop filter bandwidth, the noise from the oscillator is the same as if the VCO was running open loop. This is pointed out very clearly in *The ARRL Handbook* PLL discussion (See Note 3) and several of the other references. It is not intuitively obvious that this can be true — I doubted it until I saw the noise spectrum of my LO reduced just as the *Handbook* said!

A PLL/VCO oscillator will have a noise versus frequency spectrum that has a characteristic shape with a flattening of the noise outside the loop filter bandwidth. The flattening of the noise occurs at a difference from the carrier center frequency equal to the filter bandwidth. In an earlier version with a loop filter bandwidth of 3 kHz I could see a slight increase in noise starting at exactly 3 kHz away from the carrier. When using the LO with my down-converter I could hear an increase in noise exactly 3 kHz away from zero beat when tuning across a carrier — exactly as predicted in *EasyPLL*!

For the lowest noise, you should use the lowest N number (largest step size) possible; it's easier to filter the reference "spurs" and there is much less phase noise due to the lower N number.

When I first began using *EasyPLL*, I specified using a comparison frequency of 100 kHz. The 100 kHz channel spacing came from the original article in the *RIG Journal*. That application needed to set the LO to several frequencies that were spaced multiples of 100 kHz. This low of a frequency wasn't necessary in my application but I didn't think to change it to 2 MHz — I could have specified a higher frequency that resulted in a lower divisor for both the N and R dividers. *EasyPLL* is so smart that it suggested increasing the comparison frequency from 100 kHz to 2 MHz so that there would be less noise in the carrier — that's pretty smart software! (Remember that any noise in the reference is multiplied by 20 log N, where N is the divide ratio.) In switching from 100 kHz to 2 MHz, the noise multiplication dropped by a very significant 46 dB. (For a comparison frequency of 100 kHz, the noise multiplication is 20 Log 225600 = 107 dB but for a comparison frequency of 2 MHz the noise multiplication is only 20 Log 1128 = 61 dB.)

You should download the datasheet for the LMX2326 PLL IC.[14] Review all the information in it along with the other references.

Physical Layout

I took extra care when designing the power distribution portion of the circuit. The goal was to prevent coupling signals from one stage to another over the power lines. To accomplish this, I used one regulator each for the PLL, VCO and PIC. There are lots of bypass capacitors on the power lines to provide each IC with a very good ac ground on the power leads. Figure 4 shows the circuit board pattern that I created. An etching-pattern version of this image is available on the *QEX* Web site.[15]

The power supply and Vtune lines are kept physically separated from the RF output — you don't want any RF getting on these lines. The RF output has a low impedance, and the Vtune input has a very high input impedance (it is typically a varicap) — as a result the Vtune is by far the most susceptible to stray input signals. The pin-outs of the PLL and the VCO are arranged such that the power supply and Vtune lines must cross each other at some point. I used an RF choke to feed dc to the VCO while keeping the Vtune trace short.

Note that the path from the PLL to the VCO consists of very little circuit board trace and mostly loop filter components — the path should be as short as possible so there is no "antenna" to pick up stray signals.

Figure 3 — Parts List

Capacitors	*Value / Description*	*DigiKey p/n, other mfgr*	*Price*
C1	5-25 pf miniature variable	see text	
C2, C4, C6, C13, C15, C22	100 pF chip size 1206	478-1480-1-ND	$4.18/10
C10	0.001 µF chip size 1206	478-1530-1-ND	$3.03/10
C3, C5, C8, C14, C18, C20,C23, C24	0.01 µF chip size 1206	478-1542-1-ND	$3.03/10
C7, C9, C11, C12, C21	0.1 µF chip size 1206	478-1556-1-ND	$2.20/10
C16	4.7 pF chip size 1206	478-1464-1-ND	$3.52/10
C17	680 pF chip size 1206	478-1490-1-ND	$5.13/10
C19	180 pF chip size 1206	478-1483-1-ND	$5.13/10
C25	33 µF 16 V epoxy dipped tantalum	478-1894-ND	$1.75
Resistors			
R1, R3	2.2 MΩ chip size 1206	RHM-2.2M-ECT-ND	$.91/10
R2, R4, R5	1 kΩ chip size 1206	RHM-1K-ECT-ND	$.91/10
R6, R7, R14, R15, R16	18 Ωm chip size 1206	RHM-18-ECT-ND	$.91/10
R8	10 kΩ chip size 1206	RHM-10K-ECT-ND	$.91/10
R9, R11	180 Ω chip size 1206	RHM-180-ECT-ND	$.91/10
R10	30 Ω chip size 1206	RHM-30-ECT-ND	$.91/10
R12	3.9 kΩ size 1206	RHM-3.9K-ECT-ND	$.91/10
R13	10 kΩ size 1206	RHM-10K-ECT-ND	$.91/10
Transistors			
Q1, Q2	2N5460 JFET, P chan, TO-92	2N5460OS-ND	$0.46
ICs			
U1, U3, U5	78L05 5 V reg TO-92	LM78L05ACZNS-ND	$0.73
U2	LMX2326 PLL IC	LMX2326TM-ND	$3.20
U4	PIC16F84 PIC, 4 MHz, 18 pin DIP	PIC16F84-04/P-ND	$6.13
U6	2250-2300 MHz VCO	CRO2275A see text	$29.95 In quantities of 3
Misc			
RFC	15 µH. RF choke	M1376-ND	$0.30
X1	10.000 MHz fundamental	ICM 35UAAFF21OG	$38.35
	18 pin DIP IC socket w/flat solder tabs		see text
J1	SMA female connector	ACX-1230-ND	$4.37

QX0801-Kocs04

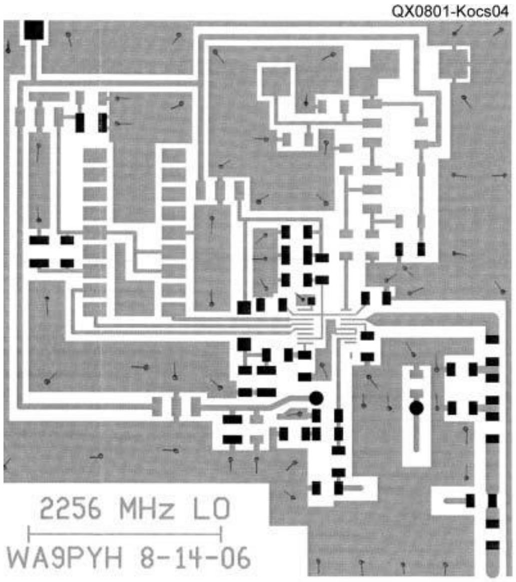


Figure 4 — This circuit board pattern shows the location of the wires used to connect the circuit ground traces with the ground foil on the bottom of the circuit board. A copy of the circuit board etching pattern is available on the *QEX* Web site. See Note 15.

WA9PYH

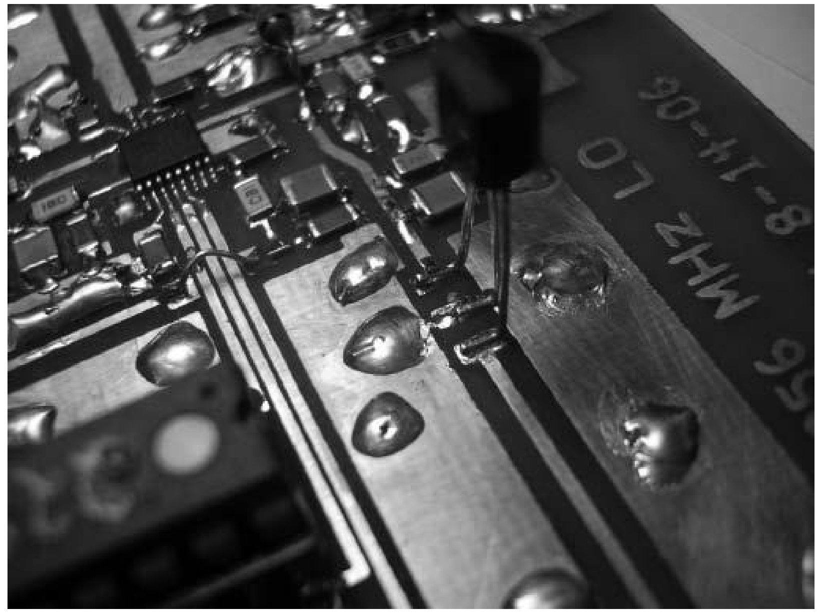


Figure 5 — This photo shows how the leads of a regulator are bent to solder the "feet" to the circuit board traces. The solder "blobs" are covering ground wire connections to the bottom of the board.

Construction

The lead photo shows the completed board with all parts installed. Take your time installing the components. They are very small, and it is very easy to ruin the PC board. Use bright lights and a magnifier so that you can see what you're doing clearly. Move parts into place slowly and skip coffee for a day. You'll need good vision and steady hands when soldering the PLL IC and VCO.

Plated through holes for grounds would have been nice but since there is no simple way to make them at home, I use another technique. At the 59 locations where a ground is needed, I drilled a small hole (0.020 inch diameter), inserted a ¼ inch long piece of no. 26 wire, and soldered both sides to connect that point to ground. See Figure 4 for the locations and the direction each wire must face. The wires must be bent going the directions shown so they don't interfere with component placement at some locations. (Drilling those holes took about an hour using a small drill press.) The line on the board that is located between my call sign and the date is exactly 1 inch long, and is used to check that the artwork scaling is exactly 1:1.

All components are surface mount except for the voltage regulator ICs, RF choke, PIC socket, crystal, trimmer capacitor and the FETs. You will need to bend their leads on those components to fit the pads. Figure 5 is a close-up view of one of the regulators mounted to the board.

A socket is shown for the PIC. There's no problem with lead length here, plus you can reprogram the device if you want to change frequency (within range of the VCO, of course).

The soldering iron should have a grounded tip with a 3 prong plug so there is no chance of a static discharge into any of the parts. The board should be connected to the same ground during assembly. You should use a grounded wrist strap so your body doesn't inject static electricity into the delicate parts. I have found a Weller WP-25 to be a suitable soldering iron, and I use a Highland 2272 grounded wrist strap.[16] These are available from Digi-Key. Suitable soldering flux and desoldering braid are also listed in Note 15.

If you've never worked with surface mount parts, practice on a scrap board until you're comfortable positioning and soldering them. I use the following technique: Use a small sponge soaked in water to wipe the soldering iron tip frequently, place a small amount of liquid soldering flux where the part attaches to the board, place the component on the board near where it goes, slide it into place using a toothpick, hold it in place with the toothpick, put a very small amount of solder on the tip, solder one side, let it cool, then solder the other side. Use just enough solder to secure the component. Examine a commercially made board containing surface mount (SMT) parts to see the correct amount of solder.

A 0.060 to 0.070 inch flat blade tip will work fine for all parts except the ground plane connections. A wider tip (0.125 inch wide blade) is a good size to use for soldering the ground feed-through connections on the ground side of the board. The ground side is a big heat sink that can cool a small tip quickly resulting in a poorly soldered connection, so use a large tip.

The board must be clean and free of all solder flux when you're done, especially in the loop filter and RF areas. Any remaining flux may change the properties of the loop filter and short some of the RF signal to ground. Spray flux remover can be used but its vapors can be rather toxic. I use rubbing alcohol and a small brush to remove most of the flux, then do a final cleaning with the spray flux remover.

Sequence of Parts Installation

1) Put a small amount of soldering flux where each component or grounding wire is soldered to the board.

2) Install the 59 grounding wires through all holes. Rotate each jumper into approximately the correct position as shown in Figure 4. Place a small amount of flux on the wire and ground plane side of the board, solder the ground plane side first, then rotate the circuit side of the wire into the required position, place a small amount of flux on the circuit side and solder that side. Cut both ends of the wire with an X-Acto knife so the wire doesn't extend to where it can short to another trace or catch on any surface.

3) Install the jumper wire that crosses over the three traces that go between the PLL and PIC. See Figure 6.

4) Install the LMX2326 PLL IC. Pin 1 is marked by a faint small dot on top of the IC as shown in the Data Sheet — make sure that you have it positioned properly! There is a small unconnected "dash" on the circuit

board located to the right and slightly above where the PLL IC is located. That "dash" should line up with the pins on that side of the IC. This will ensure that you have the PLL IC located properly. Position the IC pins *exactly* over the 16 pads. Hold it in place with a toothpick and solder one of the corner pins. Do this by touching the iron tip to the trace, and *not* the IC pin. Allow the solder to wick up to the IC pin from the trace. Solder the pin at the opposite corner in the same manner. If the IC has moved and is not positioned properly, unsolder it and try again. Minimize the time heat is applied to the IC. If there's any chance you have overheated the IC, replace it, because they are very inexpensive. If all has gone well so far, solder the remaining pins by allowing the solder to wick up to the remaining IC leads. If solder bridges occur between the pins, apply some liquid flux to the spot and use desoldering braid to remove the extra solder. I did this successfully several times.

5) Install all resistors and capacitors except those near the VCO (R9-16, C17-22 and the RF choke). (Note that there are no pads for C25 but its location is shown in Figure 6. It was added after I noticed that I had failed to include a large valued filter capacitor on the +12 V input to the crystal reference oscillator. Adding it reduced the noise spectrum outside the loop bandwidth by 6 dB.)

6) Install the VCO. The VCO is installed *after* the PLL so that if you mess up soldering the PLL and cannot recover your mistakes, you only lose the time you spent, the cost of the board and a few dollars in parts. The VCO is the second most expensive ($30) and most difficult-to-remove part. Make sure the solder wicks up into the cavities for the power, V_{Tune}, RF output and ground. See Figure 7. Also see Note 12, AN-107 "How to solder Z-Comm VCOs" and AN-100/1 "Mounting and Grounding of VCOs."

7) Bend the PIC IC socket pins outward so they lie flat on the circuit board. Trim them if they extend beyond the solder pads where the socket is mounted. Solder all pins to the board.

8) Install each regulator by first bending ⅛ inch of all three leads forward, then solder only the input and ground leads (the right and center leads when looking at the flat side). Their orientation is shown in Figure 6. Power up the board and verify that 5 V appears at each unconnected output lead. (I installed the regulators backwards and fried the PLL and a wide-range VCO — that was a very expensive and time-consuming mistake.) Once you've verified that 5 V dc appears at the three output leads, solder those leads in place.

9) Install the two FETs. Their orientation is also shown in Figure 6. Note that they face opposite directions.

10) Install the remaining components near the VCO (see Step 4, above).

11) Install the crystal and trimmer capacitor.

12) Install the output connector. See Figure 8.

13) Install the PIC into its socket.

Power-Up and Check-Out

1) Apply 12 V dc power. The circuit as shown should draw approximately 43 mA.

2) At power-up, the VCO will drift for 5 or 6 seconds, then snap to 2256 MHz. The tuning voltage at the VCO V_{Tune} pin should read very close to 1.46 V dc when measured with a high impedance DMM.

3) Attach a frequency counter to the Q2 drain. Adjust C1 until it reads exactly 10.000000 MHz.

4) If at all possible, view the output on a spectrum analyzer to verify output purity. If you have access to an RF power meter or a spectrum analyzer that can indicate absolute power in dBm, verify that the output level is approximately –2 dBm.

5) The individual data bits can be seen on pin 18 of the PIC, the clock for loading each data bit on pin 1 and the Load Enable (LE) on pin 17. Note that there will be 21 or 20 pulses on pins 17 and 18, but only 1 pulse on the LE pin for each of the N, R and F registers. The reason for this is that the LE pin loads 20 or 21 bits into the selected register and does this all at once.

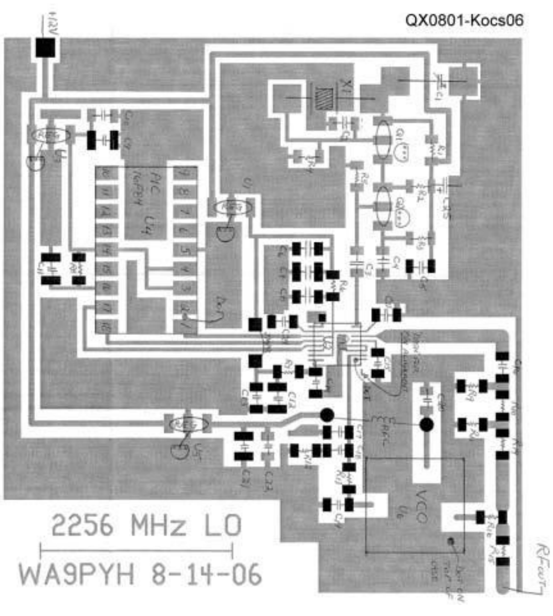


Figure 6 — This parts-placement diagram shows the location and orientation of the various components. All of the components use surface mount.

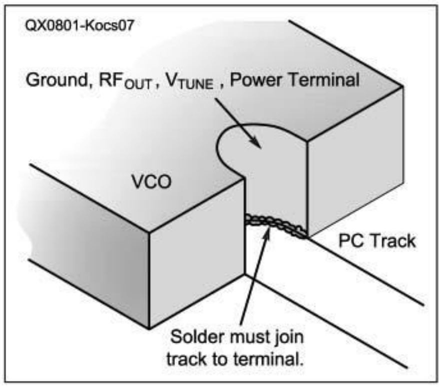


Figure 7 — This drawing illustrates the mounting technique used for the three connections to the CRO2275 Z-Communications VCO.

About the Parts

The crystal must be a fundamental type. If you use a third overtone type of crystal, the circuit will oscillate at ⅓ the frequency marked on the crystal case. This will make finding a crystal a bit difficult. For my frequency, a third overtone 30 MHz crystal is needed, and one was not available at hamfests. The part specified from ICM is a special low phase noise fundamental type crystal. It is a bit expensive ($47) but will produce a very clean 10 MHz signal when used in the FET circuit. I tried some really cheap crystals (from a PC plug-in board) in the oscillator circuit and at their natural frequency they were very noisy when viewed on a spectrum analyzer.

I bought three VCOs from Z-Comm even though their "policy" is to sell a minimum of five. I talked to an applications engineer (Ajay) to plead my case to buy only three (one for the board as shown, one for the down-converter in final configuration and one spare). He was very helpful and arranged for me to buy the three I needed. If you want only one for yourself, go in with some fel-

WA9PYH

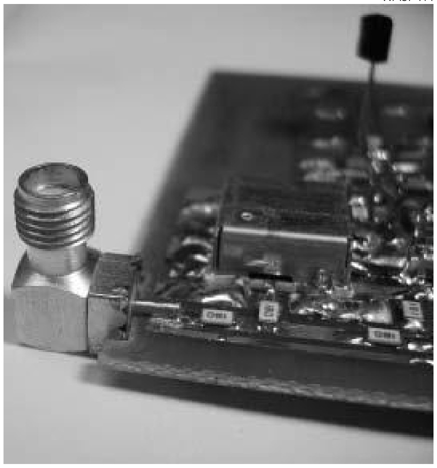


Figure 8 — The SMA output connector solders to the edge of the circuit board.

Computing the R and N Register Values for a 10 MHz Reference And 2256 MHz Output

Sections 1.7.5 and 1.7.6 of the LMX2326 datasheet have more detailed information about these calculations.

I selected a PLL comparison frequency of 2 MHz for my LO, so the R value is 5 and the N value is 1128. (Note 10 MHz / 5 = 2 MHz and 2256 MHz / 1128 = 2 MHz.)

Computing the bit pattern for R

The R register is loaded into the LMX2326 by sending a 21 bit serial data stream. The last two bits, C1 and C2, are called Control Bits. They specify the register being sent the data bits. To load data into the R register they are both set to 0. The PIC software picks off individual bits from three 8-bit bytes (the bytes are the values 0, 12 and 128 in the program listing). The first and second 8-bit bytes are sent in their entirety. Only the first 5 bits of the third byte are sent. Figure S1 shows the significance of each bit and the order in which they are sent.

The actual R value is located in bits R1 through R14 in binary form. The weight of each bit is shown for the entire 14-bit R number. The weight of each bit in each 8-bit byte is also shown. The value of 5 (10 MHz reference frequency divided by 5 = 2 MHz comparison frequency) is set by turning on bits R1 and R3.

To set R to a different value, compute its value in binary and turn on (set to 1) those same bits in the three bytes. You will likely use a very low number so that only the first or second byte will be different from mine. For example, if your reference is 20 MHz and you want a 2 MHz comparison frequency, set R to 10, by turning on the following bits: R2 and R4. All other bits will be off (set to 0).

Computing the bit pattern for N

The N register is loaded into the LMX2326 in the same way as the R register, but requires that you perform a series of calculations first. The N register consists of 21 bits, the last two bits are the Control Bits (as in R) but for the N register they are set as follows: C2 = 0 and C1 = 1.

In my local oscillator, N is set to 1128. This value was determined by the LO output frequency and the comparison frequency: 2256 MHz / 2 MHz = 1128. There are two counters/dividers in the "N" section of the LMX2326. They are called the A "swallow" counter and the B counter. Together they divide by 1128. See the calculation examples below and the example provided in section 1.7.6 of the LMX2326 datasheet. In all three examples P = 32 — it is a fixed value in the LMX2326.

Example 1 — My Circuit

Step 1: Assume N = 1128, P = 32, Frequency out is 2256 MHz, the comparison frequency is 2 MHz.

Step 2: Set B = N / P = N / 32 = 1128 / 32 = 35.25. Drop the decimal portion and set B = 35.

Step 3: Set A = N – (B × P) = N – (B × 32) = 1128 – (35 × 32) = 1128 – 1120 = 8. So, A = 8.

In summary, the A "swallow" counter is set to 8 and the B counter is set to 35.

Note that the A counter is set with bits N1 through N5 and the B counter is set with bits N6 through N18, and they are spread across all three 8-bit bytes.

The bit pattern for N = 1128 is shown in Figure S2.

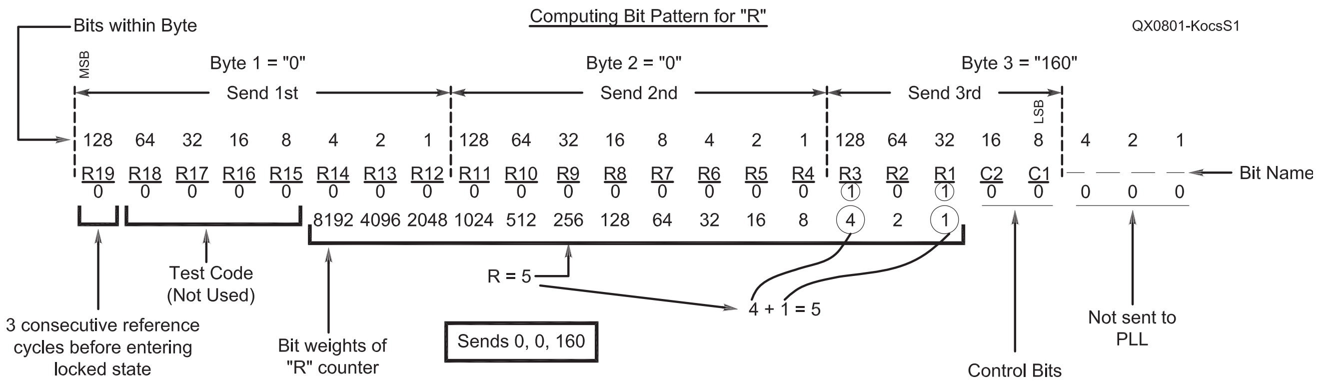


Figure S1 — Here is a chart showing the method used to determine the bit pattern to be loaded into the LMX2326 R register.

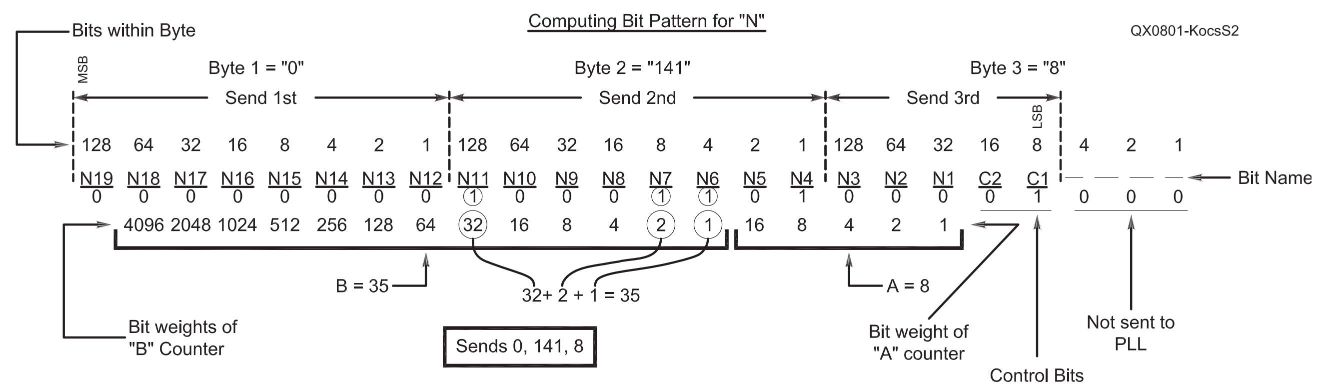


Figure S2 — This chart shows how to compute the bit pattern to be loaded into the LMX2326 N register.

WA9PYH

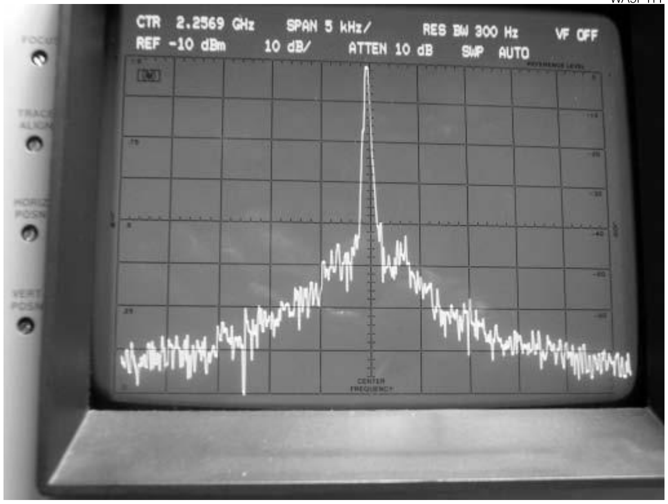


Figure 9 — This photo of the spectrum analyzer display shows the local oscillator output.

WA9PYH

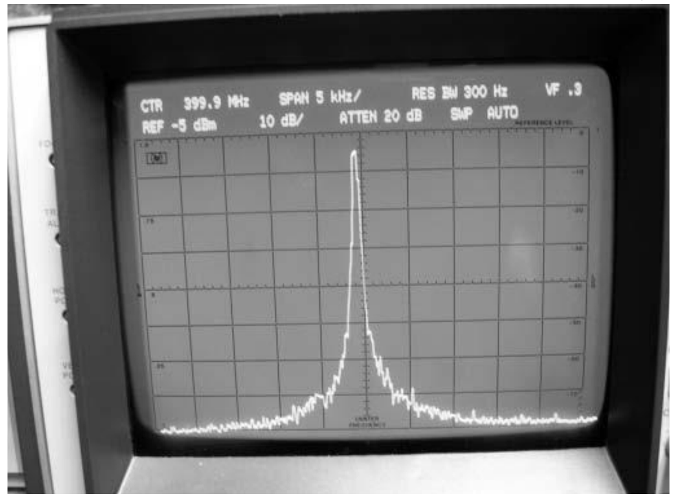


Figure 10 — For comparison with the LO spectrum display shown in Figure 9, this display shows the output from an HP 8640 signal generator set to 400 MHz.

low hams and buy a small quantity. The datasheet for this VCO is not available on their Web site, but it is available on the *QEX* Web site. (See Note 15.) The VCO I used has been replaced by a new lead free part. It is a CRO2275LF. Z-Comm is following the new industry Reduction of Hazardous Substance (RoHS) standard of having no lead in components. Electrically, the new part is the same as the "leaded" version and will solder into place using 60/40 or 63/37 solder.

I'm not in the business of selling programmed PICs but if readers have no other way to obtain a programmed PIC, I can provide them already programmed for this frequency only. Please send $8 for each PIC — US addresses only.

The PIC IC socket is one that I found at a hamfest. I couldn't determine if those available from Digi-Key had pins that are flat and can be bent out toward the sides of the socket body, so look for these in your junk box and at hamfests. Only one 18 pin socket is needed, and should cost less than $0.50.

The pads for the 0.1 µF chip capacitors are wider than the pads for the other capacitors. This is because the 0.1 µF capacitors I had on hand were wider than those shown in the parts list. The type shown in the parts list will fit the board, and actually have room to spare.

C1, the 25 pF variable capacitor used to tune the reference oscillator, is one that I found in my junk box. It measures ⅜ inch in diameter, with tabs that are bent to fit the pads on the circuit board. I was not able to find any suitable units at Digi-Key. Hopefully you have one in your junk box or can find one at a hamfest.

J1, the output connector, also came from a hamfest and costs around $0.50.

Software

The PIC program file (in assembly language — human readable) and the HEX file (loaded into the PIC) are on the *QEX* Web site. (See Note 16.) Look through the assembly program to find the loops that strip off and send one data bit at a time, the clock and load enable (LE) signals to the PLL IC. It is very well commented, and after spending a little time examining the program flow you will see how the bit patterns are generated. Note that the PIC software uses the "Counter Reset Method" described in paragraph 1.7.4 of the LMX2326 data sheet. The book I used to learn PIC programming and to understand this program is *Easy PIC'n* by David Benson.[17] [Note that this is not Dave Benson, K1SWL, of Small Wonder Labs. — *Ed*.]

The Sidebars show how to calculate the R and N register values and set up an account to use *EasyPLL* on the National Semiconductor Web site.

Results

The bandwidth I entered into *EasyPLL* for the version shown in this article was 10 kHz. Unfortunately, I don't see that same dip in the noise on the spectrum analyzer. See Figure 9. Also the noise isn't as low as predicted by *EasyPLL*. I'm attributing some of the differences to not having plated through holes in my design. Perhaps the filter needs

WA9PYH

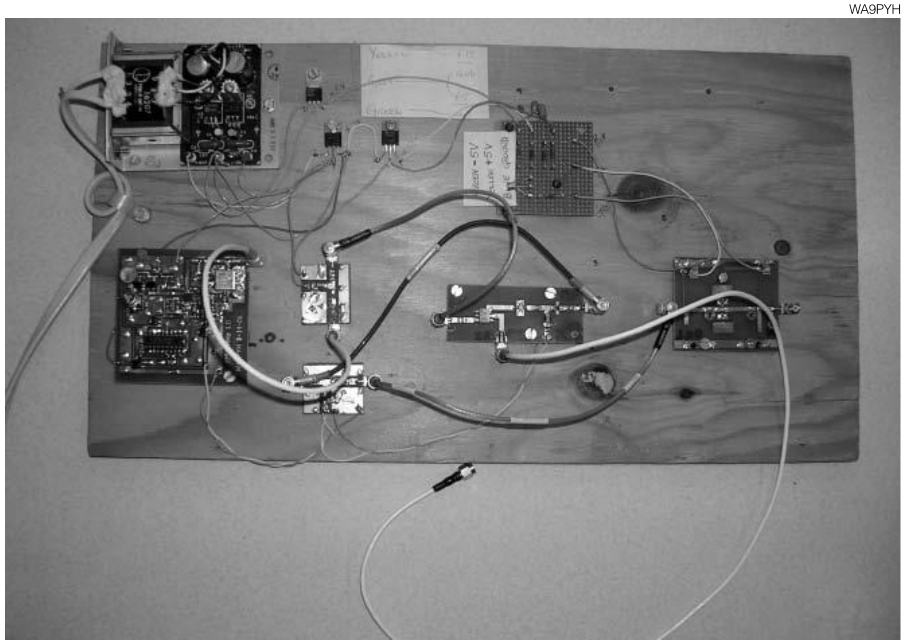


Figure 11 — This 2400 MHz downconverter for satellite operation uses the 2256 MHz LO, shown in the lower left corner of the "breadboard."

Setup and Using *EasyPLL*

First you must open an account on the National Semiconductor Web site:

1) Connect to **www.national.com**.

2) Click "Wireless" under the "Design" heading.

3) On the right side, under "My Designs," click "sign-on here."

4) Click on "here" in "You may create your own personal work space on National's Web site here."

5) Fill out all the information: e-mail address, password, re-enter password, name, address and so on.

6) Click "create."

From now on, the Web site will recognize you (on the same PC) or log on from any other PC using your e-mail address and password.

The Web site provides access to all your previously stored designs, plus you can update a design and save it under a new name.

Using EasyPLL

1) Connect to **www.national.com**.

2) Under "Design" click on "Wireless."

3) Click on "start here."

4) Select the following:

Loop filter type: Passive Loop Filter

PLL System Specifications:

Min output freq 2256 MHz

Max output freq 2256 MHz

Channel spacing 2000 kHz

Crystal Freq 10.0 MHz

Max Power Supply 5.5 V

PLL Selection Options:

Check only: single PLLs, Integer PLLs, Automatically Narrow PLL Choices, Voltage Check.

5) Click "Now View Recommended Parts."

6) Select LMX2326 for the PLL.

7) Select "any" for the VCO, and then edit the data:

Freq min 2250 MHz

Freq max 2310 MHz

Gain 15 MHz/V

Tune volts min 1.5 V

Tune volts max 4.5 V

Pout 0 dBc

Power Supply 5.0 V

VCO cap 50 pF

10 kHz noise floor –114 dBc

(All this data is available on the Z-Comm Web site for the CRO2275A VCO. Note: For the frequency tuning voltage values, use the "Typical tuning curve" graph on page 2 of the data sheet, not the numbers listed in the table on page 1.)

8) Click "Create a design." (A block diagram and many fields appear.)

9) At this point each parameter has a HELP icon that you can click. Each help file provides a lot of important information on that parameter. I suggest printing out each of the HELP files (use Shift-Print Screen, then paste the clipboard into "Paint" or other graphics software, then print it out). As a "beginner," I was constantly opening the HELP files — it is more efficient to have all the HELP files available at once.

10) Modify/verify the values in "Advanced Settings."

Uncheck "allow 2 parallel capacitors"

Filter order 3rd order

Resistor Tolerance 10 %

Capacitor Tolerance 10 %

11) Modify/verify the values in "Loop Filter Specification" as follows:

Comparison freq 2000 kHz

Charge pump gain 1.25 mA — see Note 18.

VCO gain 15 MHz/V

VCO input cap 50 pF

Output freq 2256 MHz

Phase margin 50° 0%

Loop Bandwidth 10 kHz 20%

T3/T1 Ratio 40% 20%

12) Modify/verify the values in "Loop Filter Optimization."

Spur Offset freq 2000 kHz

Initial freq 2256 MHz

Ending freq 2256 MHz

Tolerance 1000 Hz

Optimize for Min High Order Cap

Max lock time 30,000 µs

Max spur gain 100,000 dB

Max High Order Cap 50 pF

13) Scroll to the top of the page and verify all values to make sure none have changed. Print the entire page.

14) Click "View Recommended Component Values." This is when the software does its magic and computes the values of the loop filter resistors and capacitors. In about 20 seconds the screen updates, showing the values. Print the entire page. At the upper right, save your design by clicking "Rename Design." Enter a name that will identify it, then enter any additional information in the "comments" box.

15) Click "Analyze a Design."

16) Under "Phase Noise" click "Go."

17) The curves show the phase noise contribution of various components.

18) Select various traces to see the noise contribution of each component. Close this window when you're done examining the predicted noise spectrum.

19) Click "Build It."

20) Click on "Design Check."

21) Note the Design Check parameters, the result and comments for each. All show "OK" except for three parameters.

A) "Capacitor Dielectric Check" — This is telling/reminding you to use loop filter capacitors that have a dielectric that has a very nearly or zero temperature coefficient. Types NP0/C0G are preferred. Unfortunately, Digi-Key does not offer a 0.01 µF for C18 with an NP0/C0G dielectric. I used what I had in my parts box.

B) "Discrete Sample Effects Check" – I don't understand the comments for this parameter. However, I know that lock time is not a concern in this design so this can probably be ignored.

C) "Low Filter Order Check" — the spur levels can be reduced but since I could not even see them on the spectrum analyzer I ignored this comment. Also, if using a higher order filter, more parts will be required.

further design work. The capacitor that I used for C18 is a type that the Design Check part of *EasyPLL* says *not* to use. See step 21.A of the Set-up and Using *EasyPLL* Sidebar for a discussion on this issue. Trying new values in the actual circuit requires removing existing resistors and capacitors and installing new values. A circuit board can only take so much of this before the traces start to lift off the board. I used *EasyPLL* and Dean's Book until everything looked as good as possible, and then I built the circuit. As we say at work, "It's time to shoot the engineer and build the product."

Figure 10 shows the spectrum of my HP8640 signal generator at 400 MHz for comparison with the spectrum of my LO, shown in Figure 9. Note that the LO spectrum isn't as "clean" as the HP signal generator. The LO spectrum was not as clean as predicted in *EasyPLL*, but I believe is good enough to use for satellite work.

I constructed a down-converter for 2400 MHz satellite operation that is not in final form as this is written — it is a true breadboard, and is shown in Figure 11. I compared the sensitivity of my down-converter to a commercial down-converter (a model AIDC 3731 from K5GNA) with a 1 dB noise figure. I found that they had the same sensitivity when receiving a very weak signal (–127 dBm). My down-converter has no input filter to remove the noise from the image frequency (LO – IF = 2256 MHz – 144 MHz = 2112 MHz in this case). When adding the filter, the noise figure may be even lower!

The LO in the K5GNA down-converter is much cleaner than my LO, and slightly cleaner that the HP8640. Apparently, mine is adequate for weak signal reception.

Summary

I learned a *lot* developing this project. Getting to see the output on a spectrum analyzer is a real treat. Better yet, when using a slow clock frequency for the PIC, you get to see the VCO free run and drift, then snap (lock on) to 2256 MHz when the PLL is loaded with the data to make it run.

I talked with the author of the *RIG Journal* down-converter article (see Note 2) via e-mail and he provided lots of help. We used digital pictures of my layout and the spectrum analyzer, voltage measurements, and so on to discuss problems I encountered. As a result, we became good friends. He's not a ham but is a very dedicated and capable homebrewer. He lives in England.

I strongly urge you to read all the literature you can find on PLLs — in particular *The ARRL Handbook*, Mini-Circuits' *VCO Designers Help Notes* on their Web site, and most importantly "Dean's Book." PLLs are a different beast but you *can* learn their operation. If you don't understand something, contact the manufacturer and the authors of the articles. Do a Web search on "RF PLL Design." I'll help as much as possible via e-mail and *snail-mail*.

This design can be used for other frequencies as well. This will require changing the PIC variables, loop filter values and the VCO. Z-Comm and Champion are two manufacturers of narrow range VCOs in a package style that will fit this board layout. While Mini-Circuits makes VCOs that will fit this board, they don't make a narrow range VCO in the range I needed for this project. The PLL IC operates from 100 MHz to 2800 MHz, so you can build an oscillator for any frequency in this range using the correct VCO type, reference oscillator frequency, loop filter values and constants in the PIC program.

The level is too low for most mixers (–2 dBm). Most require +7 dBm or more, so I use a MAR-6 to increase the output level for proper injection level.

The PIC clock is set by R8 and C11. The values shown produce a very low clock frequency — about 85 Hz. I used a low frequency clock for the PIC clock so that I can monitor the three lines going to the PLL IC. I can literally watch the code being loaded into the PLL! It takes six seconds to load the data and start up the PLL. I haven't tried a higher frequency but plan to do so later.

Please send any questions or comments to me by e-mail or US Mail. I am open to suggestions to improve the design. If you choose to build this LO, I would be very interested in hearing from you. Good luck — I hope you learn as much as I did!

Notes

[1]Jim Kocsis, WA9PYH, "A Synthesized Down-Converter for 1691-MHz WEFAX," *QEX* Mar/Apr 2000, pp 48-55.

[2]*Remote Imaging Group Journal*, No. 65, June 2001, pp 62-81.

[3]Mark Wilson, K1RO, *Ed.*, *The ARRL Handbook*, PLL section, Chapter 10, "Oscillators and Synthesizers," 2008 Edition, pp 10.33-10.46. *The ARRL Handbook* is available from your local ARRL dealer, or from the ARRL Bookstore, ARRL order no. 1018. Telephone toll-free in the US 888-277-5289, or call 860-594-0355, fax 860-594-0303; **www.arrl.org/shop**; **pubsales@arrl.org**.

[4]**www.minicircuits.com/pages/app_notes.html** Scroll down to "VCOs."

[5]**www.national.com/appinfo/wireless/files/deansbook4.pdf**.

[6]"Design a PLL for a Specific Loop Bandwidth," *EDN*, Oct 12, 2000.

[7]"Design Loop Filters for PLL Frequency Synthesizers," *Microwaves & RF*, Sep 1999.

[8]"PLL Synthesizers," *EDN*, Mar 14, 1997.

[9]"Model PLL Dynamics & Phase Noise Performance," *Microwaves & RF*, May 2000.

[10]"Dealing with PLL Generated Phase Noise," *RF Design*, August 1997.

[11]"An Analysis & Performance Evaluation — A Passive Design Technique for Charge Pump PLLs," National Semiconductor Application Note AN1001.

[12]**www.zcomm.com/support/application_notes.shtml**.

[13]John Clark, K2AOP, "A Simple, Well-Behaved Crystal Oscillator," Technical Correspondence, *QST*, Sep 2004, p 67. Also see changes and corrections in *QST*, Oct 2004, p 39 and Dec 2004, p 37.

[14]**www.national.com/ds/lom/lm2306.pdf**. The page will open in Adobe Acrobat. Click "save as" and store it on your PC. Although the file is named "2306," it covers the LMX2326 as well.

[15]The files associated with this article are available for download from the *QEX* Web site. Go to **www.arrl.org/qexfiles** and look for the file **1x08_Kocsis.zip**.

[16]A suitable soldering iron is a Weller WP-25, Digi-Key WP-25-ND $35.99, with tips ST1-ND $4.35 and ST4-ND $4.35, a suitable grounded wrist strap is Highland 2272, Digi-Key SCP-172-ND, $7.46, Chemtronics desoldering braid, Digi-Key 80-2-5-ND, $3.20 and Kester type SP-44 soldering flux, Digi-Key KE-1700-ND, $1.28. See **www.digikey.com**.

[17]David Benson, *Easy PIC'n*, Square 1 Electronics, PO Box 1414 Hayden, ID 83835; **www.sq-1.com.** This book is out of print. A new version, *Easy Microcontrol'n* is available from Jameco — **www.jameco.com**. Order p/n 215061B for $32.75.

[18]The LMX2326 is capable of being set to two charge pump levels: 1.25 mA and 1.00 mA. Setting it to 1.00 mA allows lower values for the loop filter resistors. These lower values result in less noise contributed by the resistors. The LMX2326 is usually run at 3 V but if run at 5 V, the charge pump levels increase by 25% resulting in a charge pump level of 1.25 mA (1.00 × 125% = 1.25).

Jim Kocsis, WA9PYH, has been a ham since 1964. He earned his Amateur Extra class license in 1986. His interests include manually sent CW, PSK, casual DXing, ragchewing on HF, QRP CW Field Day operating and homebrewing just about any ham radio equipment. He has designed and built a downconverter and dish for 1691 MHz weather satellite reception, a 436 MHz circularly polarized Yagi for satellite work, and is currently building a downconverter for 2400 MHz satellite work. He has also built an azimuth and elevation rotator system that interfaces with SatPC32. He makes his own circuit boards, including the board for this article, but they are limited to SMT components. Jim is a Test Engineer for Honeywell Aerospace, with more than 30 years of experience. Some of Jim's other interests include cooking and baking, reading travel books, traveling and noncompetitive bicycling.

QEX