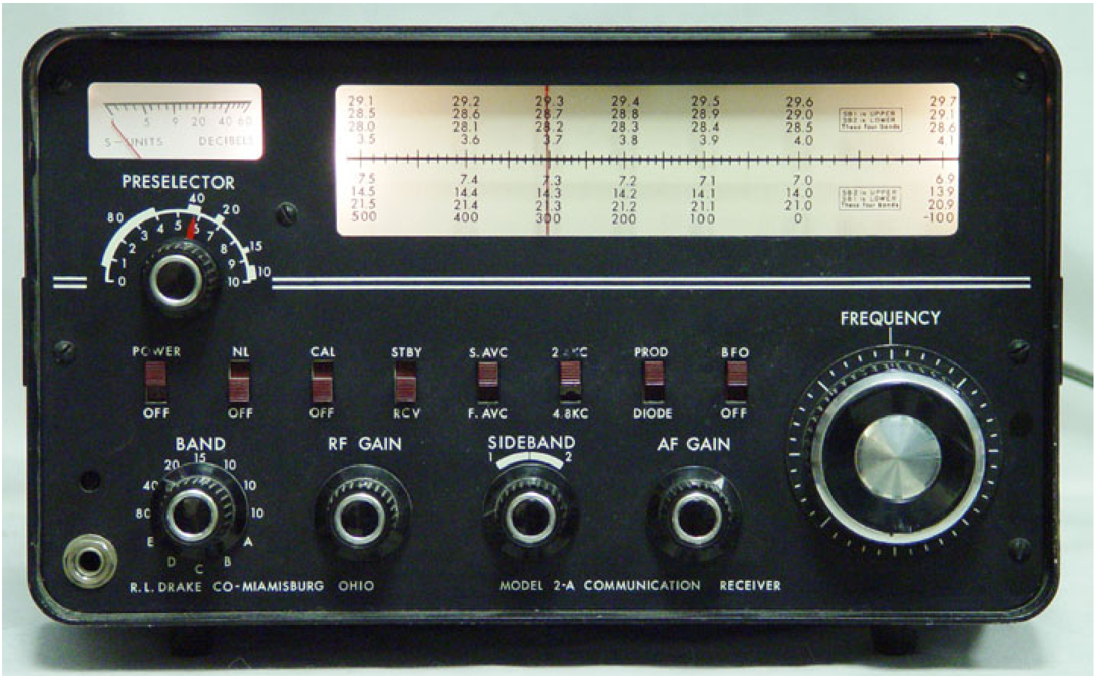
K7QO Marker Generator

The history of marker generators begins with the commercial receivers of the early beginnings of electronics. Typical short wave receivers came with two dials, one labeled tuning and the other labeled bandspread.

The typical receiver covered from 500kHz to 30MHz in band ranges selected by a front panel switch. The ham bands were shown on the dial display as extra wide lines. You would set the main tuning dial to say 7.0MHz and then tune within the range of the 40 meter band using the bandspread dial.

The problem was setting the dial to the correct frequency. Especially if you were going to use the receiver with a VFO tuned transmitter. How would you determine the band limits accurately? To do this required the use of a 100kHz crystal used in an oscillator that was rich in harmonics all the way past 30MHz. The unit was usually an additional option that was purchased at the time you bought the receiver or later from a local dealer or by mail order. Remember. This was before the days of the Internet.

On the next page is a picture of the Drake 2A receiver. Many of us have owned one at one time or another. You can see the dial layout clearly in this photograph. Note the third switch from the left labeled CAL for the optional crystal calibrator. You would turn the calibrator on and tune to one of the 100kHz positions on the dial and zero beat the tone. Then you would slide the display to move the corresponding reading to match the vertical red pointer to finish the calibration sequence.



Here is a photograph of the crystal calibrator module that was an option for the Drake 2A receiver. The module plugged into its octal plug location in the back of the receiver, which was open. There was no back plate on the receiver.

The VE3DNL Marker Generator

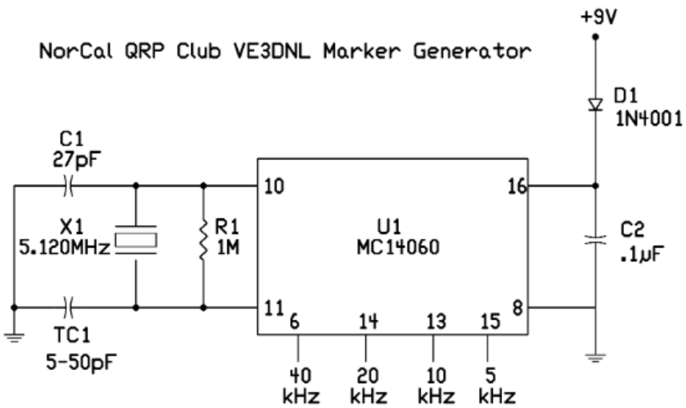
Glen Leinweber, VE3DNL, in July 1995 posted on the QRP-L mail reflector a schematic for his marker generator that used a Motorola MC14060 integrated ciruit to generate regularly spaced signals at 5, 10, 20 and 40kHz. The 14060 was a binary divider that took the crystal frequency 5.120MHz and divided it down to each of the signals given above. If you have done anything with binary numbers and computers, you recognize 512 as 2 raised to an integer power. If you divide 5.120MHz by 1024 you get 5kHz, division by 512 gets you 10kHz and so forth.

The marker generator was made as a kit by the Ft Smith AR QRP club and then later done by the NorCal QRP club. At the time there was a supply of 5.120MHz crystals readily available from several sources for reasonable pricing. That is no longer the case.

Here is a photograph of the NorCal VE3DNL marker generator from their web page. Remember the kit is no longer available.

You will note that the frequency spacing is determined by which output pad you connected a wire to. There are several ways to do this without having to have a wire soldered to a pad. You can use header pins and other mechanical systems to change the location of the wire to the outside world.

BTW, here is the schematic for the VE3DNL marker generator for reference purposes.



Also, one more important note. The supply voltage for the generator is 9V. This means that the square waves will have an amplitude of approximately 9V. This is important for determining the signal levels at each point in the entire spectrum generated. This is for those of you that already own a VE3DNL marker generator and want a comparison to the K7QO generator.

The K7QO marker generator

Early in the life of the Microchip microprocessor chips, I purchased a PICKit 1 from Microchip for the purpose of experimentation with the 12F508 8-bit processor. One of the first things I did was use the 12F508 as a marker generator. Nothing became of the project until now.

With the difficulty in finding 5.12MHz crystals, unless one wants to get a bid from a China crystal manufacturer for about $125 to get 1,000 crystals. Fine for a club project or company, but not for an individual.

I chose to use the 12F508 with a 4.000MHz crystal to generate the desired square wave forms for four frequency intervals - 5, 10, 25 and 50 kHz. These are about all one needs for a number of functions. You may ask, if you know anything about the Microchip line of processors, why I did not use the internal oscillator? The RC time constant has too much jitter in it to be comfortable to use.

The design uses one output pin of the microprocessor chip and you select the desired frequency interval with the aid of the push button. Two output pins are used to drive the LEDs to give you an indicator for the range selected.

The kit uses a coin sized battery cell for power to make the unit small. I made the prototype using a 9V battery and a 5V regulator. The owners of QRPGuys did the board layout using a 3.3V coin sized battery cell. I asked that pads be added to allow you to use an external 5.0V power source, if you want or need higher signal levels. Be sure to remove the battery or you will destroy the battery and possibly the marker generator. Also, the microprocessor will not survive a voltage source very much greater than 5.0V. You have been warned.

Since the signal is generated from a square wave, the even harmonics of the output will not be as strong as the odd harmonics of the fundamental frequencies. This effect can be used to test receiver sensitivity and to aid in peaking circuits where the adjacent odd harmonic is overloading or too loud for sensitive adjustments. Just use the even harmonic frequencies for your tests.

The following sections show some valuable uses for the generator. You can probably think of some more yourself.

Frequency Calibration

For calibrating the K7QO marker generator there are several methods.

One is to use a commercial receiver with a digital display. Tune the radio to a frequency that is a multiple of 100 kHz and turn on the marker generator and adjust the trim capacitor to zero beat with the receiver.

If you have a general coverage receiver, then tune in WWV on one of the frequencies of 2.5, 5.0, 10,0 or 15.0MHz and zero beat the signal output from the marker generator to the carrier frequency of the WWV signal. Try to use as small an antenna on the general coverage receiver in order to hear the signal from the marker generator if you can not input the signal direct from the generator to the antenna of the receiver.

This may require some experimentation on your part.

Handy RF signal source

Because the signal generator is rich in harmonics, it makes a cheap and handy RF signal source. You do not have to tune a signal generator to get a signal near the tuning range of your receiver.

Receiver adjustment for peak input response

One of the first things you usually do after completing a receiver project is the need to peak the input signal path for maximum output at the speaker. This usually involves adjusting trimmer capacitors with an RF signal input at the antenna terminal.

The nice thing about the signal generator is that you get RF signals from the VLF frequency range and into the VHF range, with the strength of the signals decreasing as you go higher in frequency. I have made some charts showing the signal strength in 2MHz increments up to 20 MHz for a reference. These are done with a TenTec 585 Paragon transceiver with a calibrated S-meter. I did the S-meter calibration with a Wavetek 3010 signal generator at the 50μV level at a number of points from 1.8MHz to 21MHz.

For adjustment of your just finished receiver you turn on the receiver and the K7QO marker generator and tune in one of the signals. Then adjust the trimmers for maximum signal response. Use the instructions provided with the receiver or transceiver to do this correctly.

One additional piece of information. If your receiver uses an even IF frequency, say 4.000MHz, 8.000MHz, 9.000MHz or whatever, there is the possibility that you will hear a faint constant tone even as you tune around the band. This is one of the signal points beating against the BFO frequency. Just ignore it in doing the adjustments.

The peaking of the receiver can be done at any time of the day or night and the band does not have to be open to get a signal.

Receiver dial calibration markings

With homebrew receivers and homebrew enclosures for non-digital displays, the marker generator is handy as an input source to determine marking placements on the front panel or dial for 5 and 10 kHz spacings and then you evenly divide the desired markings between these points.

Receiver tuning bandwidth

If you want to see how much of the frequency spectrum your homebrew receiver or kit covers, then set the marker generator to an interval of 5KHz and count the number of spots you can hear from the lowest point in the tuning range to the highest frequency you can receive.

You will have to estimate the 1 kHz intervals, but you should be able to get close.

Measure receiver drift

Because the marker generator is crystal controlled there is no drift. If you have a computer with an accurate audio frequency measuring program, you can start up the receiver from a cold start and both measure the drift and time line from the computer program. Just tune in a frequency tone near zero and then start plotting the drift from a cold start.

Oscilloscope probe alignment

Ever wonder about the trimmer capacitor adjustment on your oscilloscope probes? It is there to trim the probe to get better impedance matching between the scope input circuits and the probe.

Set the marker generator to 50kHz output intervals and input the signal into you oscilloscope. Adjust the time display on your scope to display one or two square wave intervals. Adjust the waveform height to fill the scope display.

Now adjust the trimmer cap on the probe to get the best square waveform on the display that you can. Deformation in the waveform is seen as small oscillations at the constant voltage levels at the peak and minimum of the waveform. You will easily catch on to the effect as you adjust the trimmer.

You can also see you response time of your scope as the output from the Microchip microprocessor is just about the best square wave you can get. This is good news and this is bad news. The bad news is that a square wave with a 50% duty cycle has no even harmonics.

I can generate the code in the microprocessor to move the duty cycle but because of the button polling the higher frequency intervals would have to be eliminated.

Signal Strength Measurements

The following charts are made using the K7QO marker generator and a TenTec Paragon transceiver. The S-meter was calibrated using a Wavtek 3010 signal generator and several S-9 generators from Elecraft and the NorCal club projects. These charts are for showing relative signal strength between adjacent signals at the frequencies shown and show the low end of the most popular ham bands.

Note. There is a tradeoff in the frequency interval output signal and the signal strength. For the highest strength signal use 100 kHz output. As you decrease the frequency interval value the signal strength will decrease.

Think of it this way. You have so much energy generated by the amplitude of the wave, say 3.3V when using the battery cell. That energy is divided up into all the signals. Because you have more signal points for the smaller frequency intervals the signal strength will be reduced for each point. You will see this effect in the following tables. Some receivers may not have the sensitivity to easily hear the smaller levels. Helps to determine receiver response.

An S9 signal strength on the S-meter is from the response to a 50μV RMS signal on the antenna terminal input.

The numbers in the tables are what one can expect on a calibrated receiver, but these numbers are not be construed as the final universal numbers for these measurements. They are meant to be used as a guideline only.

50kHz 3.3V			
frequency	S-meter	frequency	S-meter
20.000Mhz	S4.5	20.050Mhz	S7.0
18.000Mhz	S5.0	18.050Mhz	S7.5
16.000Mhz	S5.2	16.050Mhz	S7.8
14.000Mhz	S5.5	14.050Mhz	S8.0
12.000Mhz	S6.0	12.050Mhz	S8.8
10.000Mhz	S6.0	10.050Mhz	S9.0
8.000Mhz	S6.2	8.050Mhz	3dB/S9
7.000Mhz	S6.5	7.050Mhz	5dB/S9
6.000Mhz	S6.5	6.050Mhz	8dB/S9
4.000Mhz	S6.9	4.050Mhz	11dB/S9
2.000Mhz	S6.9	2.050Mhz	21dB/S9

50kHz 5.2V			
frequency	S-meter	frequency	S-meter
20.000Mhz	S2.9	20.050Mhz	S7.9
18.000Mhz	S3.2	18.050Mhz	S8.8
16.000Mhz	S4.2	16.050Mhz	S9.0
14.000Mhz	S5.5	14.050Mhz	9.5dB/S9
12.000Mhz	S5.5	12.050Mhz	9.8dB/S9
10.000Mhz	S6.0	10.050Mhz	10dB/S9
8.000Mhz	S6.0	8.050Mhz	12dB/S9
7.000Mhz	S6.4	7.050Mhz	15dB/S9
6.000Mhz	S6.5	6.050Mhz	16dB/S9
4.000Mhz	S6.5	4.050Mhz	20dB/S9
2.000Mhz	S7.0	2.050Mhz	26dB/S9

25kHz 3.3V			
frequency	S-meter	frequency	S-meter
20.000Mhz	S2.0	20.025Mhz	S4.3
18.000Mhz	S3.2	18.025Mhz	S5.1
16.000Mhz	S3.0	16.025Mhz	S4.5
14.000Mhz	S3.5	14.025Mhz	S5.5
12.000Mhz	S4.0	12.025Mhz	S7.1
10.000Mhz	S4.5	10.025Mhz	S7.5
8.000Mhz	S4.5	8.025Mhz	S7.8
7.000Mhz	S4.9	7.025Mhz	S8.0
6.000Mhz	S5.1	6.025Mhz	S8.6
4.000Mhz	S5.2	4.025Mhz	2dB/S9
2.000Mhz	S6.0	2.025Mhz	12dB/S9

25kHz 5.2V			
frequency	S-meter	frequency	S-meter
20.000Mhz	S0+	20.025Mhz	S6.8
18.000Mhz	S1.0	18.025Mhz	S7.5
16.000Mhz	S2.0	16.025Mhz	S7.9
14.000Mhz	S4.5	14.025Mhz	S8.5
12.000Mhz	S4.3	12.025Mhz	S8.9
10.000Mhz	S4.8	10.025Mhz	1dB/S9
8.000Mhz	S4.5	8.025Mhz	3dB/S9
7.000Mhz	S5.2	7.025Mhz	5dB/S9
6.000Mhz	S5.5	6.025Mhz	8dB/S9
4.000Mhz	S5.5	4.025Mhz	12dB/S9
2.000Mhz	S5.9	2.025Mhz	20dB/S9

10kHz 3.3V			
frequency	S-meter	frequency	S-meter
20.000Mhz	S1.0	20.010Mhz	S3.9
18.000Mhz	S2.2	18.010Mhz	S4.3
16.000Mhz	S1.5	16.010Mhz	S4.6
14.000Mhz	S2.6	14.010Mhz	S5.0
12.000Mhz	S2.0	12.010Mhz	S5.6
10.000Mhz	S3.2	10.010Mhz	S5.8
8.000Mhz	S3.0	8.010Mhz	S6.3
7.000Mhz	S3.5	7.010Mhz	S6.7
6.000Mhz	S3.5	6.010Mhz	S6.9
4.000Mhz	S3.5	4.010Mhz	S7.3
2.000Mhz	S4.2	2.010Mhz	S8.8

10kHz 5.2V			
frequency	S-meter	frequency	S-meter
20.000Mhz	S1.9	20.010Mhz	S5.0
18.000Mhz	S2.6	18.010Mhz	S5.8
16.000Mhz	S2.0	16.010Mhz	S6.3
14.000Mhz	S3.0	14.010Mhz	S6.5
12.000Mhz	S2.0	12.010Mhz	S6.9
10.000Mhz	S3.2	10.010Mhz	S7.2
8.000Mhz	S1.8	8.010Mhz	S7.6
7.000Mhz	S3.3	7.010Mhz	S7.9
6.000Mhz	S3.6	6.010Mhz	S8.1
4.000Mhz	S3.6	4.010Mhz	S8.8
2.000Mhz	S4.3	2.010Mhz	8dB/S9

5kHz 3.3V			
frequency	S-meter	frequency	S-meter
20.000Mhz	S1.0	20.005Mhz	S2.0
18.000Mhz	S1.5	18.005Mhz	S3.2
16.000Mhz	S1.0	16.005Mhz	S3.5
14.000Mhz	S1.5	14.005Mhz	S3.7
12.000Mhz	S1.2	12.005Mhz	S4.2
10.000Mhz	S2.0	10.005Mhz	S4.5
8.000Mhz	S1.5	8.005Mhz	S4.8
7.000Mhz	S2.0	7.005Mhz	S5.0
6.000Mhz	S2.5	6.005Mhz	S5.4
4.000Mhz	S2.5	4.005Mhz	S6.0
2.000Mhz	S3.2	2.005Mhz	S7.2

5kHz 5.2V			
frequency	S-meter	frequency	S-meter
20.000Mhz	S2.5	20.005Mhz	S4.0
18.000Mhz	S1.8	18.005Mhz	S4.3
16.000Mhz	S2.8	16.005Mhz	S4.7
14.000Mhz	S1.8	14.005Mhz	S5.0
12.000Mhz	S1.8	12.005Mhz	S5.5
10.000Mhz	S2.0	10.005Mhz	S5.9
8.000Mhz	S1.5	8.005Mhz	S6.2
7.000Mhz	S2.1	7.005Mhz	S6.5
6.000Mhz	S2.5	6.005Mhz	S6.8
4.000Mhz	S2.5	4.005Mhz	S7.3
2.000Mhz	S3.2	2.005Mhz	S8.5

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

                      00001    Title  "K7QO Marker Generator"
                      00002
                      00003 #include    <p12f508.inc>
                      00001         LIST
                      00002
                      00003 ;==========================================================================
                      00004 ; Build date : Sep 13 2016
                      00005 ;  MPASM PIC12F508 processor include
                      00006 ;
                      00007 ;  (c) Copyright 1999-2016 Microchip Technology, All rights reserved
                      00008 ;==========================================================================
                      00009
                      00158         LIST
                      00004         list   p=12f508
                      00005
  00000010            00006 FREQ     EQU   10      ;  general register for current frequency
  00000011            00007 counterh EQU   11
  00000012            00008 counterl EQU   12
                      00009
                      00010 ;   K7QO marker generator for the 12F508
                      00011 ;   pin 5 is output
                      00012 ;   pin 4 is push button for state change
                      00013 ;   must use 4MHz crystal for osc or jitter will occur
                      00014
0FFF   0001           00015   __CONFIG  _MCLRE_ON && _CP_ON && _WDT_OFF && _XT_OSC
                      00016
                      00017         org    h'0000'
0000   0000           00018         nop
0001                  00019 start
0001   0C40           00020         movlw  h'40'    ;
0002   0002           00021         OPTION
0003   0C38           00022         movlw  h'38'    ; GP0, GP1 and GP2 output and GP3 input
0004   0006           00023         TRIS   GPIO
                      00024
0005   02B0           00025         incf   FREQ,F   ;  move to the next frequency output value
0006   0210           00026         movf   FREQ,W   ;  load freq to bit bang the contents
```

```
0007   0E03             andlw  h'03'    ;  keep only the two low order bits
0008   0030             movwf  FREQ     ;  put current value back into freq

0009   01E2             addwf  PCL,F    ;  generate branch into following table of ju
000A   0A??             goto   f_50k_st
000B   0A??             goto   f_25k_st
000C   0A??             goto   f_10k_st
000D   0A??             goto   f_5k_st


000E              f_50k_st
000E   0C00             movlw  h'00'  ;  turn on top LED to show 50KHz frequency
000F   0026             movwf  GPIO
0010   0C04       f_50k   movlw  h'04'
0011   01A6             xorwf  GPIO,F
0012   0766             btfss  GPIO,3 ;  see if button depressed
0013   0A??             goto   change_freq  ;  yes, increment routine section
0014   0000             nop
0015   0000             nop
0016   0000             nop
0017   0000             nop
0018   0A??             goto   f_50k

0019              f_25k_st
0019   0C01             movlw  h'01'  ;  turn on first LED to show 25KHz frequency
001A   0026             movwf  GPIO
001B   0C04       f_25k   movlw  h'04'
001C   01A6             xorwf  GPIO,F
001D   09??             call   delay_5
001E   09??             call   delay_5
001F   0766             btfss  GPIO,3 ;  see if button depressed
0020   0A??             goto   change_freq
0021   0000             nop
0022   0000             nop
0023   0000             nop
0024   0000             nop
0025   0A??             goto   f_25k

0026              f_10k_st
0026   0C02             movlw  h'02'  ;  turn on second LED
0027   0026             movwf  GPIO
0028   0C04       f_10k   movlw  h'04'
0029   01A6             xorwf  GPIO,F
```

```
002A   09??          call   delay_10
002B   09??          call   delay_10
002C   09??          call   delay_10
002D   09??          call   delay_10
002E   0766          btfss  GPIO,3
002F   0A??          goto   change_freq
0030   0000          nop
0031   0000          nop
0032   0000          nop
0033   0000          nop
0034   0A??          goto   f_10k

0035           f_5k_st
0035   0C03          movlw  h'03'  ;  turn on both LEDs
0036   0026          movwf  GPIO
0037   0C04    f_5k    movlw  h'04'
0038   01A6          xorwf  GPIO,F
0039   09??          call   delay_10
003A   09??          call   delay_10
003B   09??          call   delay_10
003C   09??          call   delay_10
003D   09??          call   delay_10
003E   09??          call   delay_10
003F   09??          call   delay_10
0040   09??          call   delay_10
0041   09??          call   delay_10
0042   0766          btfss  GPIO,3
0043   0A??          goto   change_freq
0044   0000          nop
0045   0000          nop
0046   0000          nop
0047   0000          nop
0048   0A??          goto   f_5k

0049   09??    delay_10 call delay_5
004A   0000    delay_5  nop
004B   0804    delay_4  retlw  h'04'

004C           change_freq
004C   09??            call   delay  ;  let's debounce the switch
004D   09??            call   delay  ;  give a little more time for finger removal
004E   0A??            goto   start  ;  rerun program with new frequency
```

```
004F   02F2          00113 delay    decfsz  counterl,f
0050   0A??          00114          goto   delay
0051   02F1          00115          decfsz  counterh,f
0052   0A??          00116          goto   delay
0053   0804          00117          retlw  h’04’
                     00118
0001                 00119         end
```

:020000040000FA
:10000000000400C0200380C0600B0021002030E83
:100010003000E2010E0A190A260A350A000C2600F1
:10002000040CA60166074C0A0000000000000000056
:10003000100A010C2600040CA6014A094A096607A9
:100040004C0A00000000000000001B0A020C260001
:10005000040CA601490949094909490966074C0ADE
:10006000000000000000000000280A030C2600040C19
:10007000A6014909490949094909490949099B
:10008000490949096607 4C0A000000000000000009
:10009000370A4A09000004084F094F09010AF20211
:0800A0004F0AF1024F0A0408A7
:0400A8000008000844
:021FFE000100E0
:00000001FF