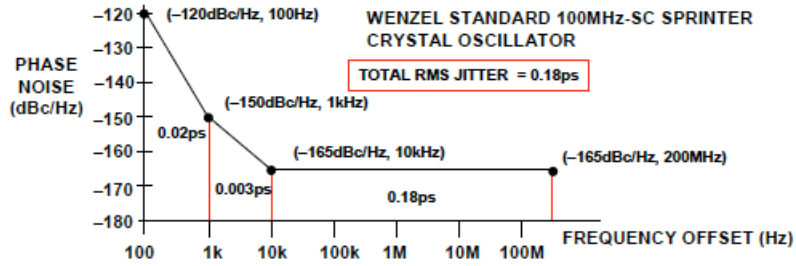
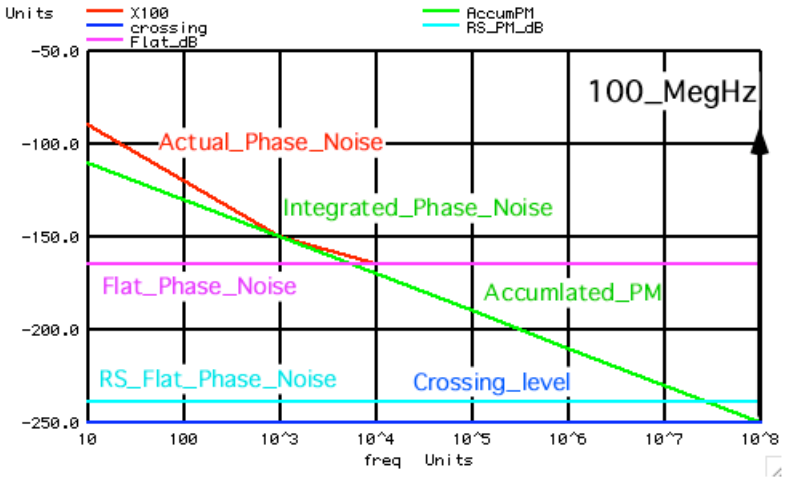
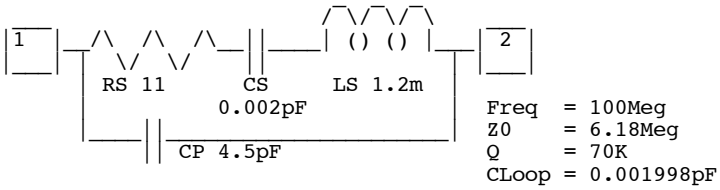
=====PHASE NOISE AT HIGHER SIGNAL LEVELS==========

So what happens when the voltage across a crystal is greater than 1V rms?
This Wenzel oscillator runs off of a +12V supply.







Now the level predicted for 1Vrms is 11dB higher than the actual crossing.
Well, 12V/2.82V is 12.6dB. So how large is the crystal voltage?

Wenzel also has provided a flat band jitter calculated value.
That value can be sanity checked by using another method.

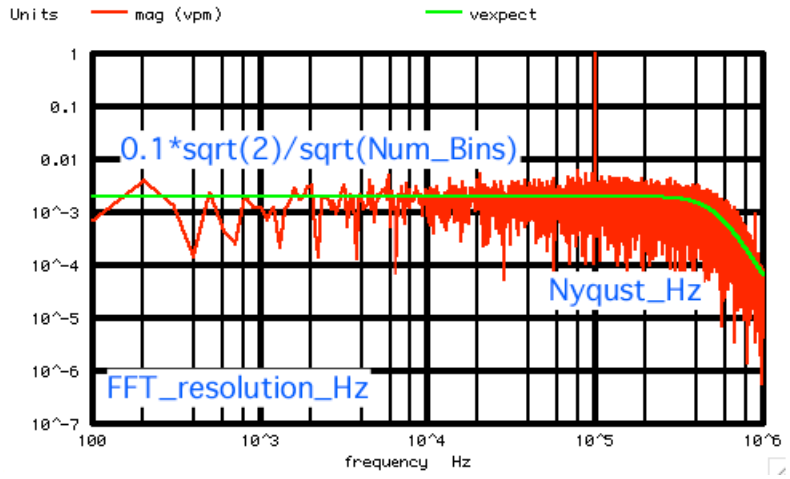
```
Circuit: Phase_Noise_100MHz_Crystal Wenzel

=======================Create_Accumulation_Slope==============
=======================Oscillator_Freq_and_Period==============
OscillatorFreq_Hz =     1E+08
Oscillator_Period_s =   1E-08
=======================Reference_Oscillator_Magnitude=====================
Osc_V_rms =             1
Osc_V_ppk =             2.82
Osc_db =                0
=======================Osillator_FlatNoise====================
```

```
Osc_Flat_Noise_V_dB =   -165
Flat_Noise_V_per_Hz =   5.62341E-09
Equivalent_Noise_R  =   1976.42
```

Mapping a one radian PM noise to a reference noise floor in terms of dB.

Ref_1Radian_magnitude = sqrt(2)/sqrt(Num_Bins)

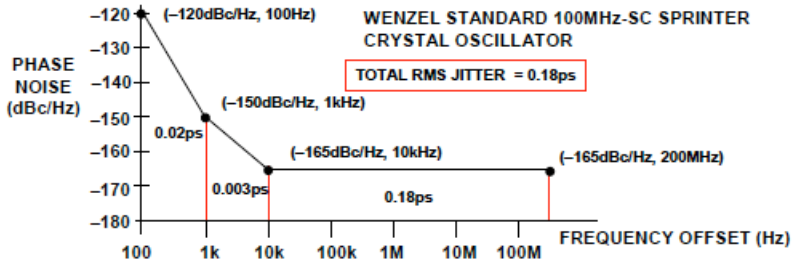



```
One_Rad_Ref_dB      =   -db(sqrt(2)*sqrt(oscFreq))
One_Rad_Ref_dB      =   -83.0103
```

The difference between -165dB and -83dB is 82dB.
That works out to 79.5u Radians rms value.

```
Jitt_floor_Rads_rms =   7.95271E-05
Flat_Jitter_rms_s   =   (period*jitt_Rad )/6.28
Flat_Jitter_rms_s   =   1.26635E-13
```

There are 2PI radians in one cycle.
Wenzel is claiming 0.18ps jitter over 200MHz.



This method is is viewing 0.127ps jitter over 100MHz.
That is 2.98 dB lower as expected.

```
=======================Timing_Tolerance_At_Osc_Freq===========
Flat_Noise_PM_dB    =   where AccumPM_crosses 100Meg
Flat_Noise_PM_dB    =   -250
=======================Listed_Q=======
Crystal_Q           =   70000
Rs_Ohm              =   11
=======================Find_RS_FlatBand_Thermal============
Therm_Noise_RS_Hz   =   4e-9*sqrt(RS/1000)
Therm_Noise_RS_Hz   =   4.19524E-10
Therm_Noise_RS_dB   =   -187.545
=======================Find_Q_Bandwidth==================
BandWidth_for_Q_Hz  =   oscFreq/Q
BandWidth_Hz        =   1428.57
=======================Find_Noise_withing Bandwidth========
Rs_Noise_in_BW_rms  =   RS_noise*sqrt(BW)
Rs_Noise_in_BW_rms  =   1.58565E-08
=======================Half_Noise_is_PM=======
Rs_PM_Noise_rms     =   RS_rms/sqrt(2)
```

```
Rs_PM_Noise_rms      =    1.12122E-08
Spread_out_over_Hz   =    1E+08
=======================Half_Noise_is_PM=======
If_Crystal_level     =    1V_rms
Expected_PM_db       =    -239.006
Flat_Noise_PM_dB     =    where AccumPM_crosses 100Meg
Flat_Noise_PM_dB     =    -250
```

Crystals do have a maximum power level.
The equation for the effective parallel resistance is given below.
Adding 10dBs signal level means 10 times more power.

```
=======================Power_at_1Vrms=======
Crystal_R_parallel   =    RS*(1+CP/CS)^2
Crystal_R_parallel   =    6.13E+06
Crystal_current      =    1.63132E-07
Crystal_Power_uW     =    0.163132
=======================done=======
```

Crystals do spec a maximum power level.
So this crystal may be operating at 1.6uW.

Table 1 • Crystal Specifications

Crystal Specifications	Min	Typical	Max	Comment
Nominal Frequency (MHz)		14.31818		
Oscillation Mode		Fundamental		
Holder Type		HC-49, HC-50		Not Important
Pin to Pin Capacitance (Co in pF)		7	10	Depends on Holder Type
Operating Temperature (°C)	–10	30	70	Application Dependent
Frequency Tolerance		± 30PPM		Application Dependent
Load Capacitance (Ceq in pF)	12.5	17	20	Affects Frequency Tolerance
Drive Level (P in µW)	0.5	1	2*	Calc. Value by (4)
Motion Resistance (Rs in Ω)	25	30	50	Affects Drive Level

* Drive level must be higher than the calculated power dissipation of the crystal as given by (4).

Wenzel Associates, Inc.
"Quietly the Best"

Crystal Oscillators > 4 to 30 MHz > **HF Ultra Low Noise OCXO**

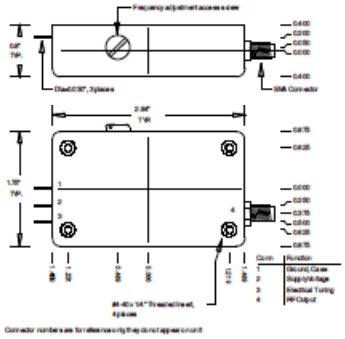
Features:

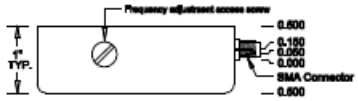
- Lowest Phase Noise Available
- Good Frequency Stability over Temperature
- Internal Voltage Regulator
- Very Low Aging Rate

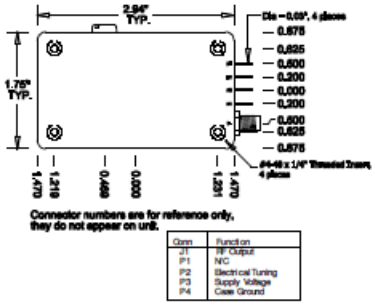
Applications:

- Radar Systems
- Reference for Phase Noise Measurements
- Synthesizers

With a noise floor available to –176 dBc/Hz at 1 kHz offset, the Ultra Low Noise OCXO has the lowest phase noise available in the industry. Measurements at NIST in Boulder, Colorado confirm phase noise performance to –180 dBc at 20 kHz for selected 5 and 10 MHz oscillators. The unit is available with temperature stability to ±1 x 10^{-8} and an aging rate of 1 x 10^{-10}, which also makes these units among the most stable. The unit has an internal voltage regulator, which provides excellent line rejection. Please consult the factory if you need any specifications to be modified to better suit your application.







Typical Specifications

Frequency (Specify)		4 to 30 MHz			
Frequency		5	10		MHz
Output Level		+13			dBm
Aging		±1 x 10^{-9} to ±1 x 10^{-10} / day after 30 days			
Phase Noise		**Std.**	**Prem.**	**Std.**	**Prem.**
	10 Hz	-145	-150	-130	-132	dBc/Hz
	100 Hz	-165	-170	-158	-162	dBc/Hz
	1 kHz	-176	-176	-174	-175	dBc/Hz
	10 kHz	-176	-176	-174	-175	dBc/Hz
Temperature Stability (Specify)					
	Range A	0 to +50C			
	Range B	0 to +65C			
	Range C	0 to +70C	±5 x 10^{-8} to ±2 x 10^{-8}		
	Range D	-20 to +70C			
	Range E	-40 to +70C			
	Range F	-55 to +85C			
Electrical Tuning Range (Specify)					
	Tuning A	0 to +10 VDC	±2 x 10^{-7} to ±2 x 10^{-6}		
	Tuning B	±5VDC			
Supply Voltage (Specify)		+12 or +15			VDC
Warm-up Power		5 for 5 Minutes			Watts
Total Power *typical*		2.5 @ 25°c			Watts
Crystal Type		SC			
Dimensions		44.4 x 74.7 x 25.4			mm
		1.75 x 2.94 x 1			inches
Connectors		SMA on side and solder pins on base			

==========MacSpiceCode==========

Phase_Noise_100MHz_Crystal Wenzel

```
*=========Create_Signal_No_Reason==================
*V_SIN#    NODE_P NODE_N DC      VALUE   SIN(    V_DC    AC_MAG FREQ    DELAY   FDamp)
VIN        VP     0      DC      0       SIN(    0       1      1                     )

.control
set       pensize = 2
unlet     stanDev_val
unlet     Out_percent
unlet     X100
unlet     freq
unlet     intnoise
let       X100              = vector(8)
let       freq             = vector(8)
let       intnoise         = vector(8)
```

```
let      offsett = -90

let      freq[0] = 10
let      X100[0] = -90
let      freq[1] = 100
let      X100[1] = -120
let      freq[2] = 1k
let      X100[2] = -150
let      freq[3] = 10k
let      X100[3] = -165
let      freq[4] = 100k
let      X100[4] = -165
let      freq[5] = 1Meg
let      X100[5] = -165
let      freq[6] = 10Meg
let      X100[6] = -165
let      freq[7] = 100Meg
let      X100[7] = -165

echo             "=======================Create_Timing_Slope=============="
let      index = 0
repeat           8
let              intnoise[index] = -20*log(freq[index]) + offsett
let index =      index +1
end

echo             "=======================Osillator_Freq_and_Period=============="
let oscFreq =    100meg
echo             "OscillatorFreq_Hz =     $&oscFreq"
let period  =    1/oscFreq
echo             "Oscillator_Period_s =   $&period"
echo             "=======================Reference_Oscillator_Magnitude====================="
let oscVppk =    2.82
let oscVrms =    1
let osc_db  =    db(oscVrms)
echo             "Osc_V_rms =             $&oscVrms"
echo             "Osc_V_ppk =             $&oscVppk"
echo             "Osc_db =                $&osc_db"
echo             "=======================Osillator_FlatNoise====================="
let Flat_dB =    X100[7]
echo             "Osc_Flat_Noise_V_dB =   $&Flat_dB"
let Flat_V  =    1/exp(ln(10)*abs(Flat_dB/20))
echo             "Flat_Noise_V_per_Hz =   $&Flat_V"
let Eq_R =        (Flat_V/4n)*(Flat_V/4n)*1k
echo             "Equivalent_Noise_R  =   $&Eq_R"

let Rad_REF_db =  -db(sqrt(2)*sqrt(oscFreq))
echo             "One_Rad_Ref_dB      =   -db(sqrt(2)*sqrt(oscFreq))"
echo             "One_Rad_Ref_dB      =   $&Rad_REF_db"

let jitt_Rad  =   1/exp(ln(10)*abs((Rad_REF_db-Flat_dB)/20))
echo             "Jitt_floor_Rads_rms =   $&jitt_Rad "

let jit_s    =   (period* jitt_Rad )/6.28
echo             "Flat_Jitter_rms_s   =   (period*jitt_Rad )/6.28"
echo             "Flat_Jitter_rms_s   =   $&jit_s"
echo             "=======================Timing_Tolerance_At_Osc_Freq============"
let expectN =    -20*log(oscFreq) + offsett
echo             "Flat_Noise_PM_dB    =   where AccumPM_crosses 100Meg "
echo             "Flat_Noise_PM_dB    =   $&expectN"
echo             "=======================Listed_Q======="
let Q =          70K
echo             "Crystal_Q           =   $&Q"
let RS =          11
echo             "Rs_Ohm              =   $&RS"
echo             "=======================Find_RS_FlatBand_Thermal============"
let RS_noise =   4e-9*sqrt(RS/1000)
echo             "Therm_Noise_RS_Hz   =   4e-9*sqrt(RS/1000)"
echo             "Therm_Noise_RS_Hz   =   $&RS_noise"
let RS_nois_dB =  db(RS_noise)
echo             "Therm_Noise_RS_dB   =   $&RS_nois_dB"
echo             "=======================Find_Q_Bandwidth===================="
let BW =          oscFreq/Q
echo             "BandWidth_for_Q_Hz  =   oscFreq/Q"
echo             "BandWidth_Hz        =   $&BW"
echo             "=======================Find_Noise_withing Bandwidth========"
let RS_rms =      RS_noise*sqrt(BW)
echo             "Rs_Noise_in_BW_rms  =   RS_noise*sqrt(BW)"
echo             "Rs_Noise_in_BW_rms  =   $&RS_rms"
echo             "=======================Half_Noise_is_PM======="
let RS_PM_rms =   RS_rms/sqrt(2)
echo             "Rs_PM_Noise_rms     =   RS_rms/sqrt(2)"
echo             "Rs_PM_Noise_rms     =   $&RS_PM_rms"
```

```
echo             "Spread_out_over_Hz  =   $&oscFreq"
echo             "=======================Half_Noise_is_PM======="
let TimeToler =   RS_PM_rms/sqrt(oscFreq)
let RS_PM_dB =    db(TimeToler)
echo             "If_Crystal_level    =   1V_rms"
echo             "Expected_PM_db      =   $&RS_PM_dB"
echo             "Flat_Noise_PM_dB    =   where AccumPM_crosses 100Meg "
echo             "Flat_Noise_PM_dB    =   $&expectN"
echo             "=======================Power_at_1Vrms======="
let RP  =         6130K
echo             "Crystal_R_parallel  =   RS*(1+CP/CS)^2"
echo             "Crystal_R_parallel  =   $&RP"
let Icryst  =     1/RP
echo             "Crystal_current     =   $&Icryst"
let Pcryst  =     Icryst/1u
echo             "Crystal_Power_uW    =   $&Pcryst"

echo             "=======================done======="

let AccumPM =    intnoise
let crossing =   expectN
plot              X100 AccumPM crossing RS_PM_dB Flat_dB vs freq xlog
plot              X100 - AccumPM  vs freq xlog


.endc
.end

4.18.11_1.35PM
dsauersanjose@aol.com
Don Sauer
```