A Digital PSK31 Meter

- *Building a digital field strength meter for your PSK31 station*

by

George Rothbart, KF6VSG

Presentation to

ARRL and TAPR 22nd Annual

Digital Communications Conference

September 19-21, 2003 -- Hartford, Connecticut

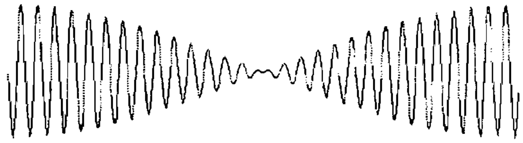
No need to take notes! This entire presentation is available at www.ssiserver.com/info/pskmeter

Digital Modes (character- based data):

- ✓CW
- ✓RTTY
- ✓Packet
- ✓APRS
- ✓SSTV
- ✓*PSK-31*

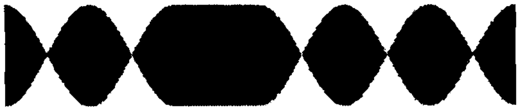
Phase shift keying—the basics

Zero bits are signaled by 180° phase reversals:



Example of a phase reversal

To avoid harmonics, the amplitude passes through zero at the instant of the reversal of phase. There is no need to change the amplitude when we don't change phase. Thus, we might see a signal amplitude like this:



PSK31 signaling

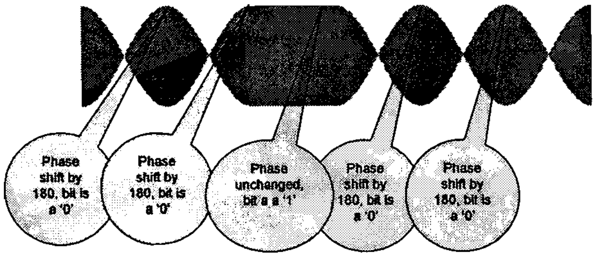
These phase changes occur every 32 milliseconds (31.25 Hz). Why this rate? Because:

- It can be easily derived from the 8KHz signal generated by your sound card (8000 Hz divided by 256 = 31.25 Hz), and
- 30 or so bits per second translates to about 50 wpm, the fastest you can type.

A zero (space) is defined as a 180 degree phase reversal from the prior sample,
A one (mark) is defined as no phase change from the prior sample.

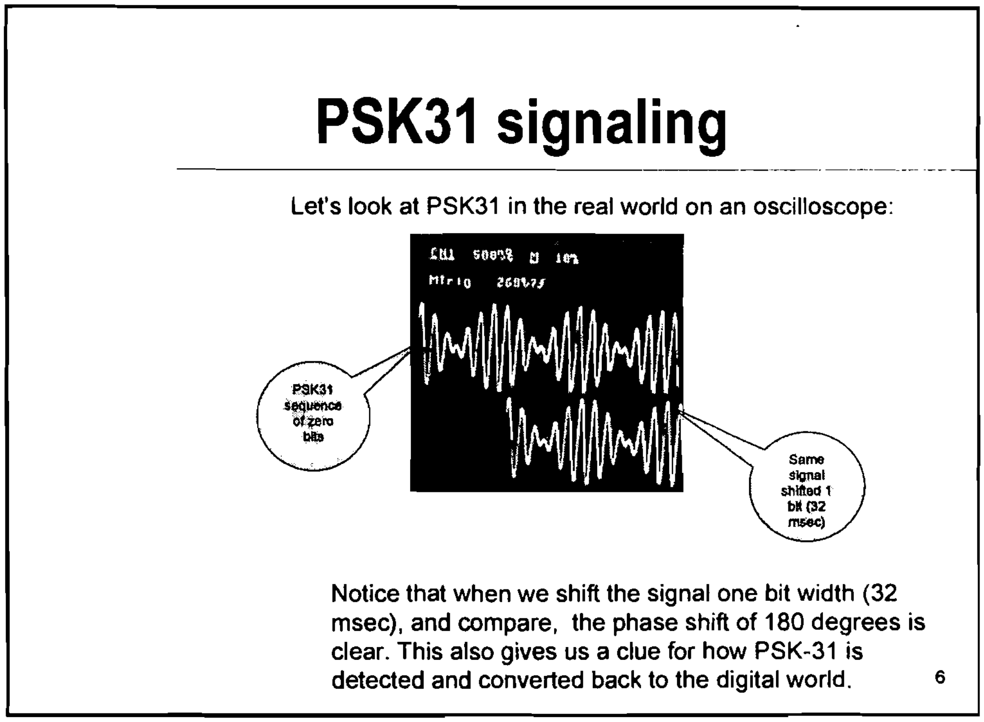
4

PSK31 signaling


Phase shift by 180, bit is a '0'
Phase shift by 180, bit is a '0'
Phase unchanged, bit is a '1'
Phase shift by 180, bit is a '0'
Phase shift by 180, bit is a '0'

Therefore, this signal is detected as 00100, which we will soon see is the representation of the character "SPACE".

5



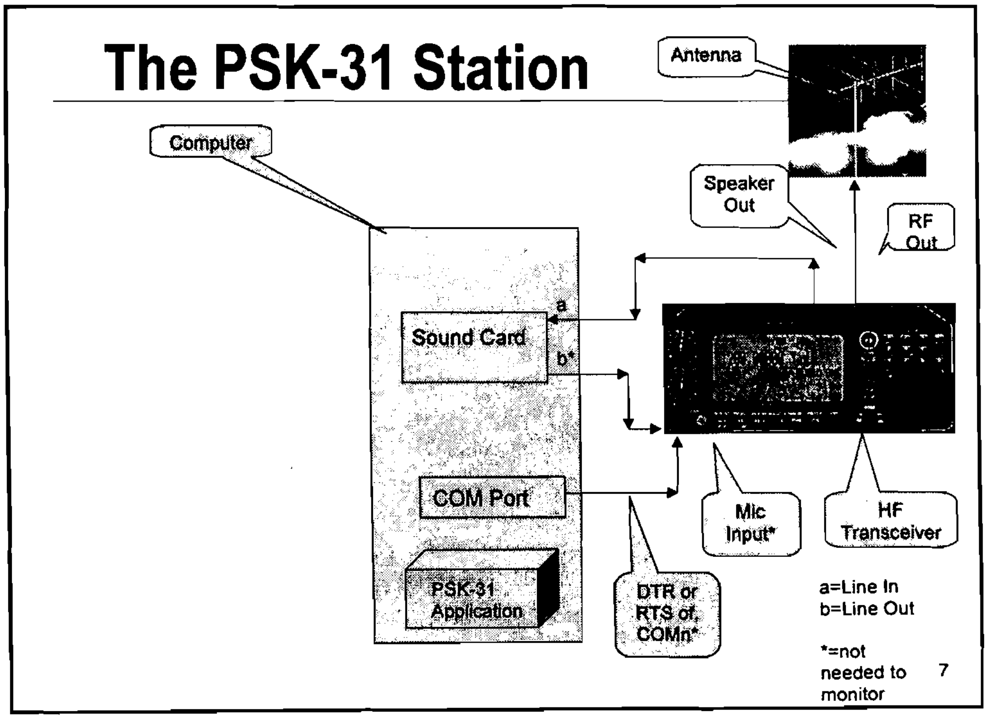
PSK31 signaling

Let's look at PSK31 in the real world on an oscilloscope:

PSK31 sequence of zero bits
Same signal shifted 1 bit (32 msec)

Notice that when we shift the signal one bit width (32 msec), and compare, the phase shift of 180 degrees is clear. This also gives us a clue for how PSK-31 is detected and converted back to the digital world.

6

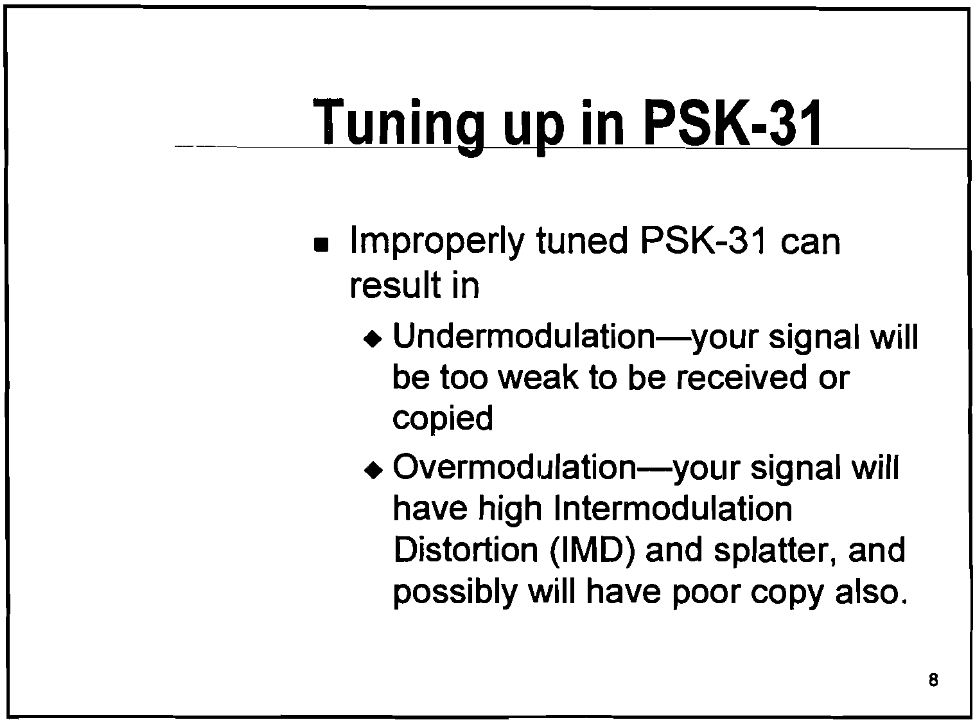


The PSK-31 Station

a=Line In
b=Line Out

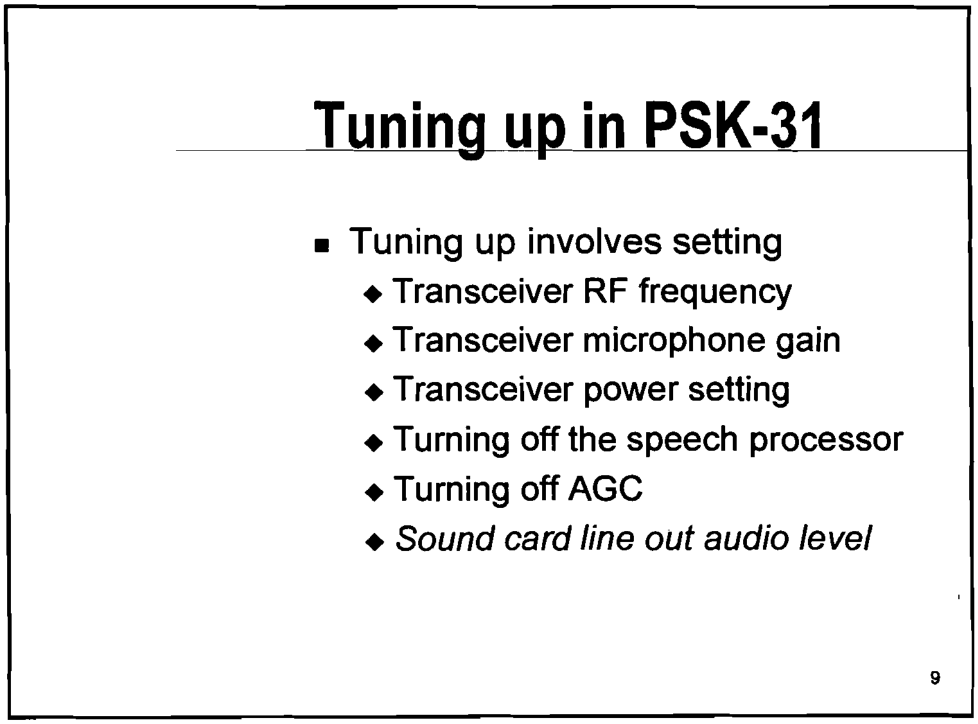
*=not needed to monitor

7



Tuning up in PSK-31

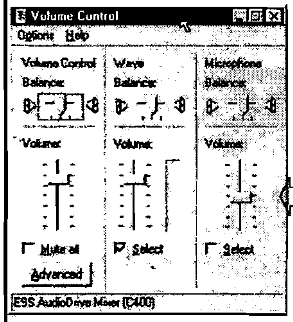
- Improperly tuned PSK-31 can result in
  - Undermodulation—your signal will be too weak to be received or copied
  - Overmodulation—your signal will have high Intermodulation Distortion (IMD) and splatter, and possibly will have poor copy also.



Tuning up in PSK-31

- Tuning up involves setting
  - Transceiver RF frequency
  - Transceiver microphone gain
  - Transceiver power setting
  - Turning off the speech processor
  - Turning off AGC
  - *Sound card line out audio level*

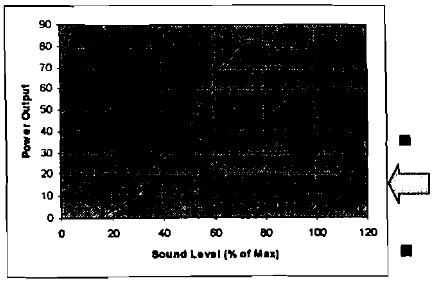
Setting Audio Levels



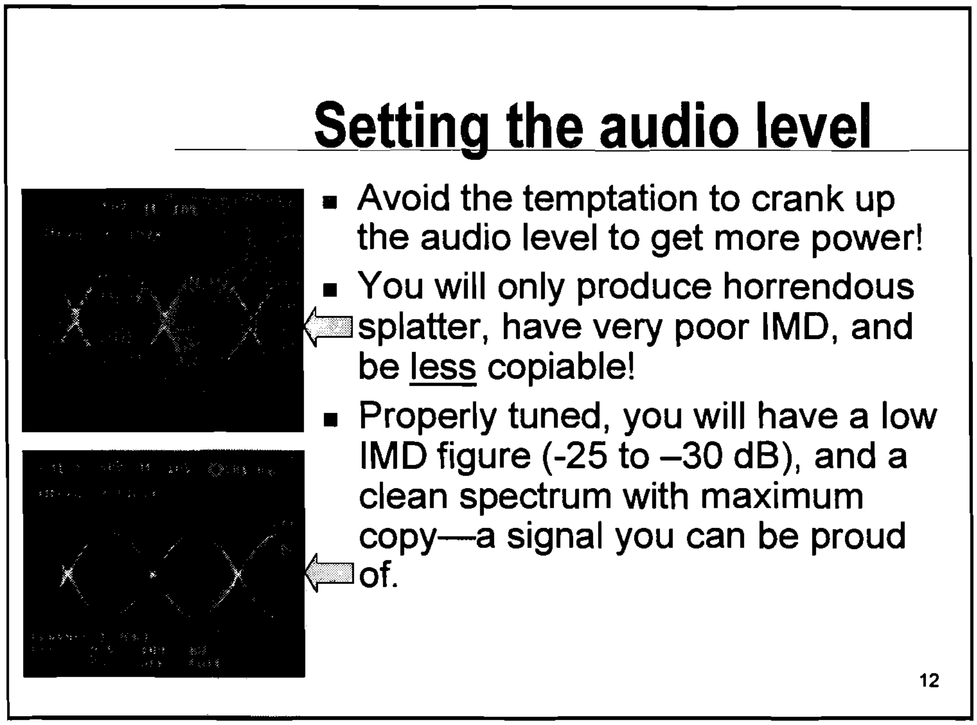
- It's all in the Window's "mixer."
- At the mixer's main display...
- De-select everything but "Wave" (that's your line out).
- Set the Wave volume to maximum.
- Now you can adjust the sound level by the Volume Control slider.

Setting the audio level

- Click "TX" button of your PSK-31 software, so that you are sending an idle PSK signal to the transceiver.
- Starting with the maximum sound level, observe the transmitter's power output and back off the mixer's audio level until the power level is at about 50%.

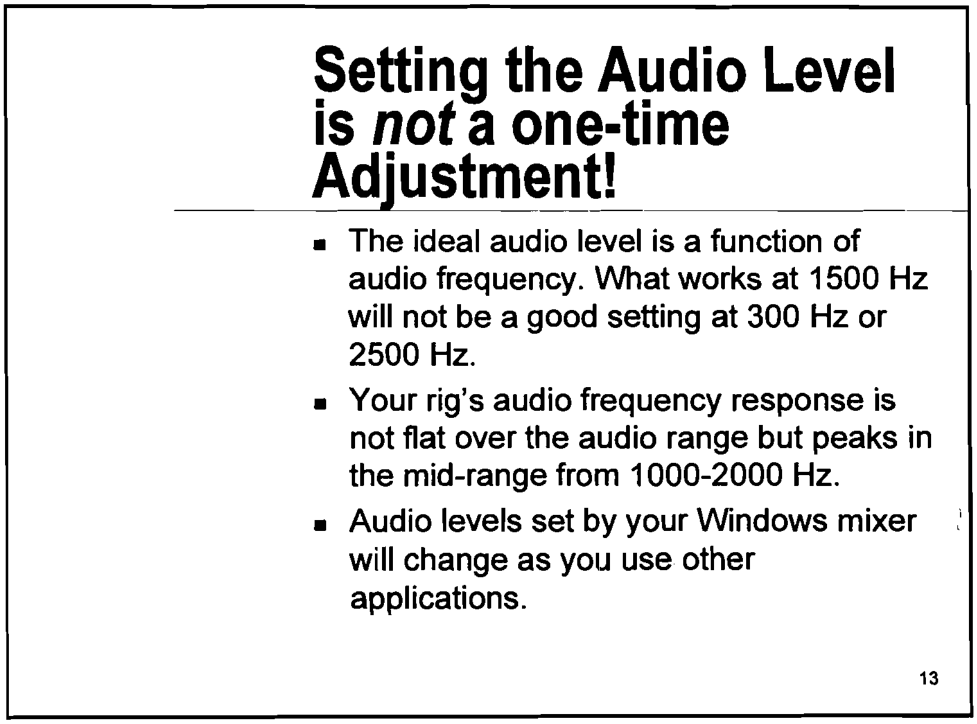


- You should observe an S-shaped curve like the one I measured at 1500 Hz.
- The proper audio level in this case would be about 50% of maximum.



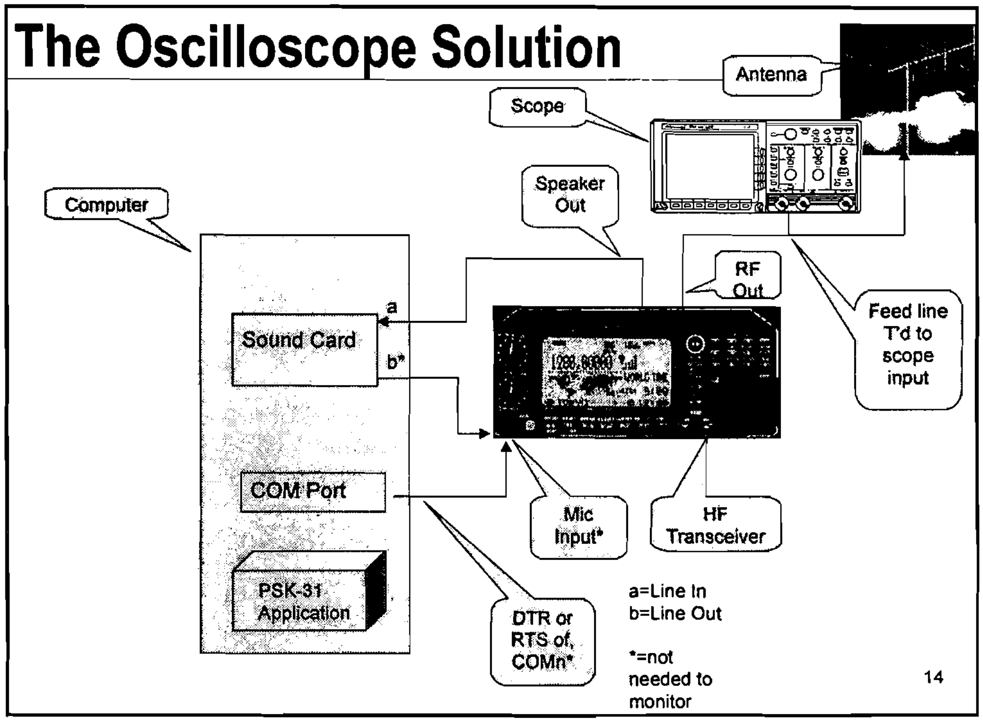
Setting the audio level

- Avoid the temptation to crank up the audio level to get more power!
- You will only produce horrendous splatter, have very poor IMD, and be <u>less</u> copiable!
- Properly tuned, you will have a low IMD figure (-25 to –30 dB), and a clean spectrum with maximum copy—a signal you can be proud of.



Setting the Audio Level is *not* a one-time Adjustment!

- The ideal audio level is a function of audio frequency. What works at 1500 Hz will not be a good setting at 300 Hz or 2500 Hz.
- Your rig's audio frequency response is not flat over the audio range but peaks in the mid-range from 1000-2000 Hz.
- Audio levels set by your Windows mixer will change as you use other applications.



The Oscilloscope Solution

Antenna

Scope

Speaker Out

Computer

RF Out

Feed line T'd to scope input

a

Sound Card

b*

COM Port

PSK-31 Application

Mic Input*

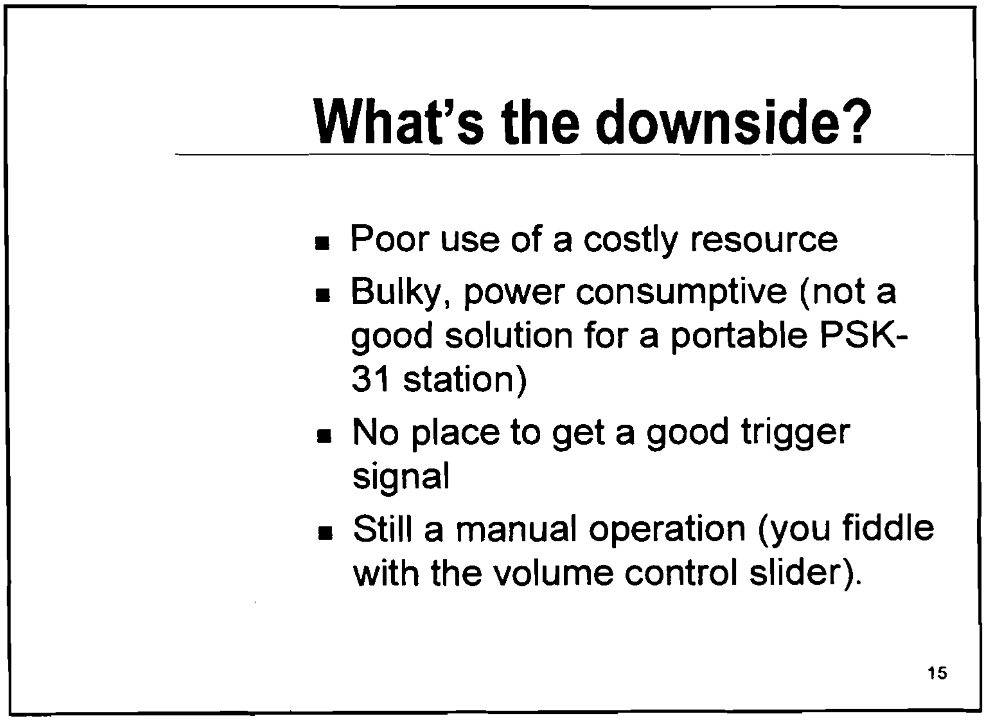
HF Transceiver

DTR or RTS of, COMn*

a=Line In
b=Line Out

*=not needed to monitor

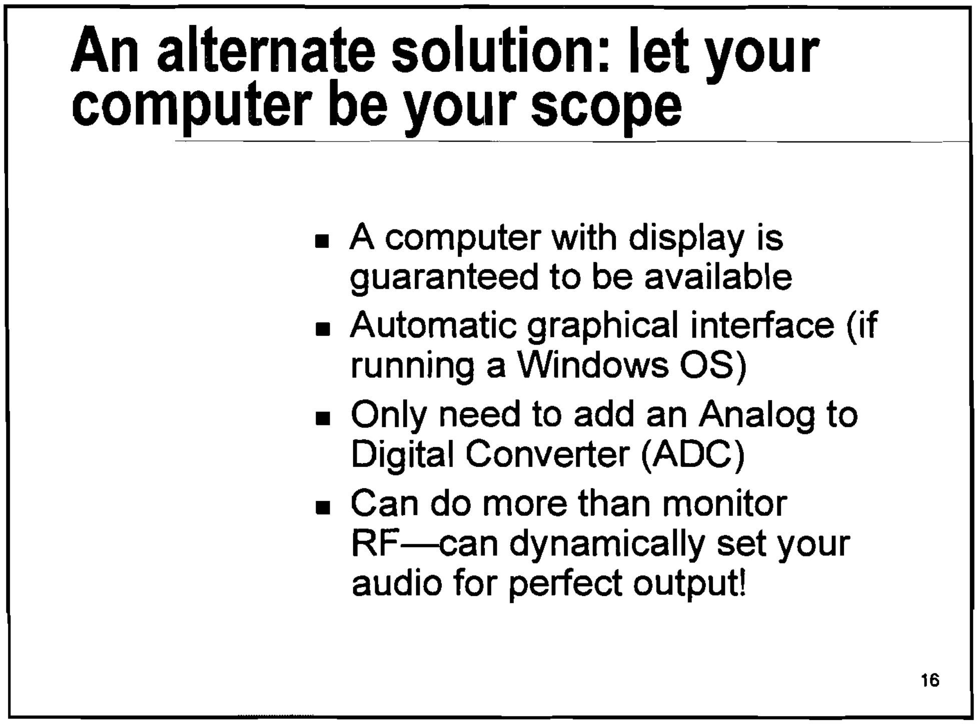
14



What's the downside?

- Poor use of a costly resource
- Bulky, power consumptive (not a good solution for a portable PSK-31 station)
- No place to get a good trigger signal
- Still a manual operation (you fiddle with the volume control slider).

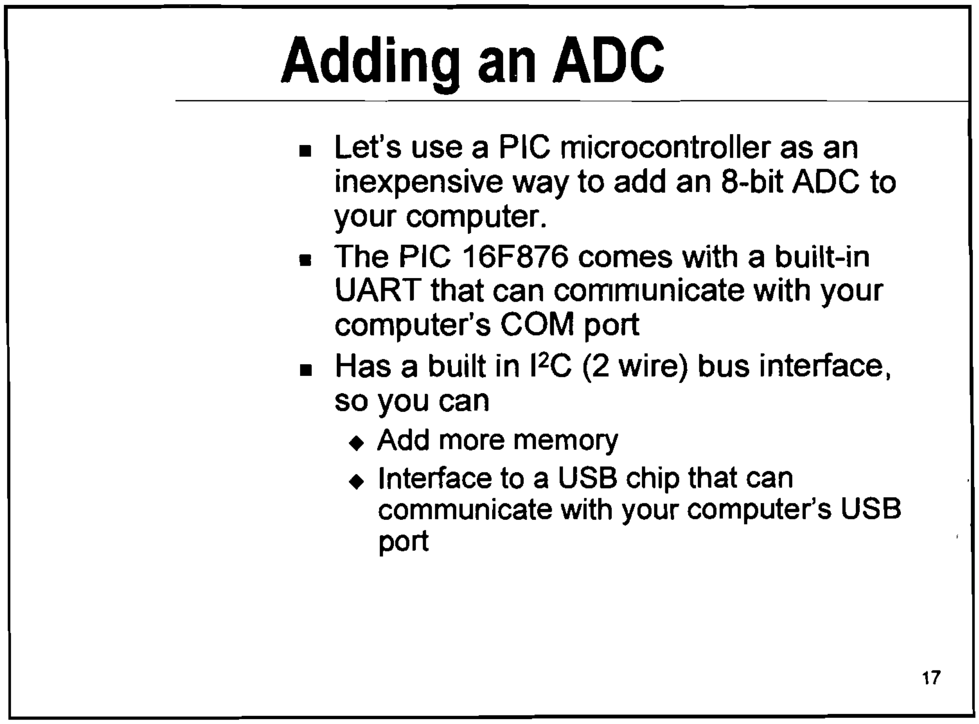
15



An alternate solution: let your computer be your scope

- A computer with display is guaranteed to be available
- Automatic graphical interface (if running a Windows OS)
- Only need to add an Analog to Digital Converter (ADC)
- Can do more than monitor RF—can dynamically set your audio for perfect output!

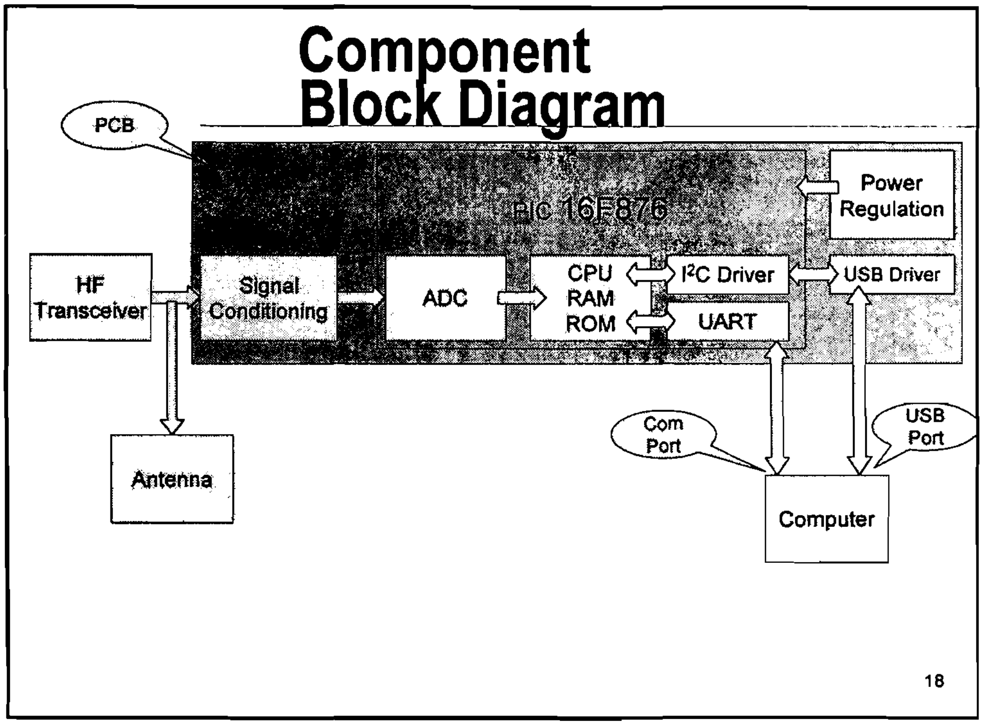
16



Adding an ADC

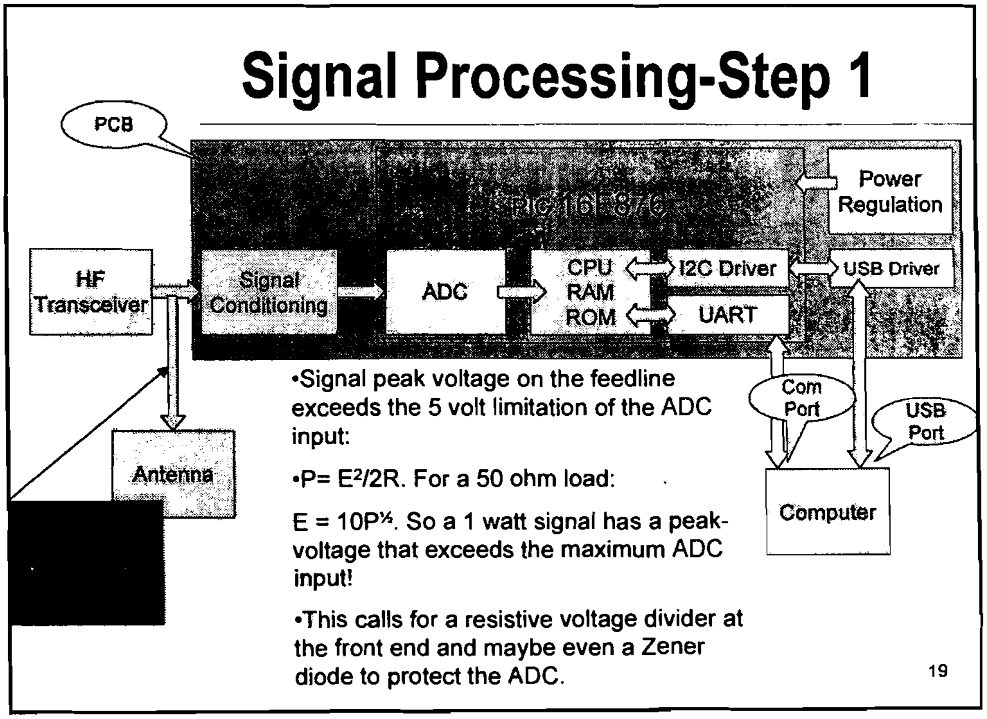
- Let's use a PIC microcontroller as an inexpensive way to add an 8-bit ADC to your computer.
- The PIC 16F876 comes with a built-in UART that can communicate with your computer's COM port
- Has a built in I^2C (2 wire) bus interface, so you can
  - Add more memory
  - Interface to a USB chip that can communicate with your computer's USB port

17



Component Block Diagram

18



Signal Processing-Step 1

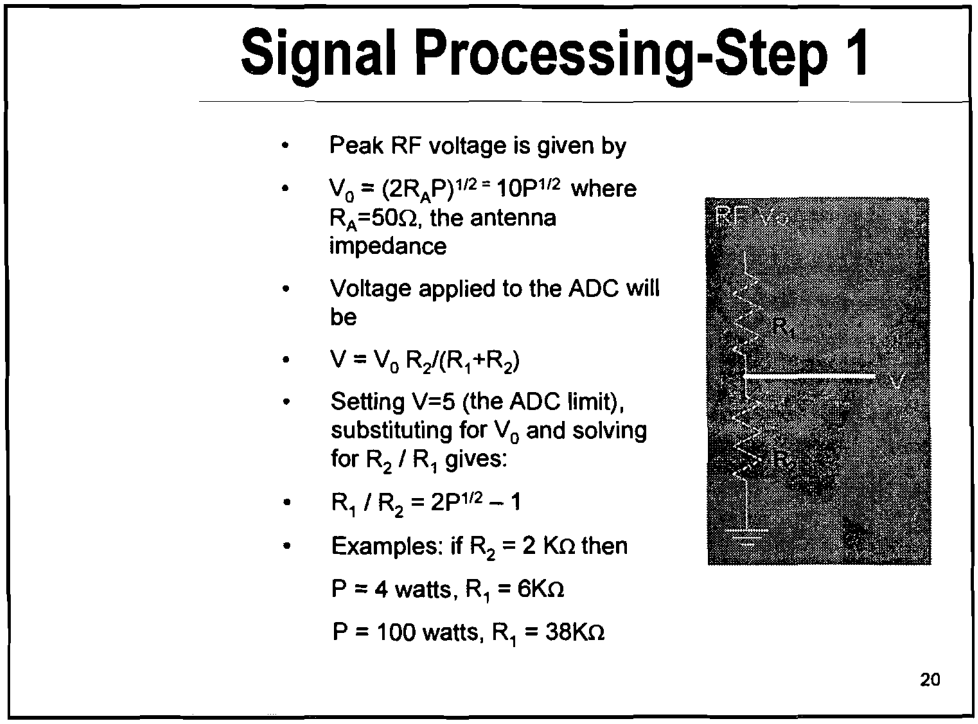
•Signal peak voltage on the feedline exceeds the 5 volt limitation of the ADC input:

•P= $E^2/2R$. For a 50 ohm load:

E = $10P^{1/2}$. So a 1 watt signal has a peak-voltage that exceeds the maximum ADC input!

•This calls for a resistive voltage divider at the front end and maybe even a Zener diode to protect the ADC.

19



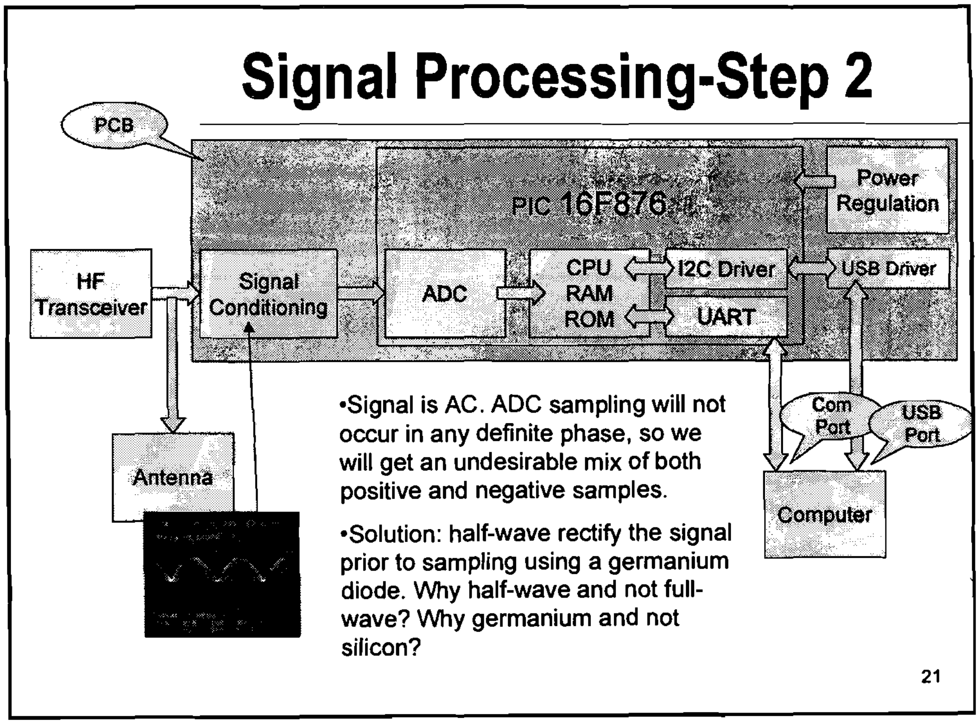
Signal Processing-Step 1

- Peak RF voltage is given by
- $V_0 = (2R_AP)^{1/2} = 10P^{1/2}$ where $R_A=50\Omega$, the antenna impedance
- Voltage applied to the ADC will be
- $V = V_0 R_2/(R_1+R_2)$
- Setting V=5 (the ADC limit), substituting for V_0 and solving for R_2 / R_1 gives:
- $R_1 / R_2 = 2P^{1/2} - 1$
- Examples: if $R_2 = 2$ KΩ then

  P = 4 watts, R_1 = 6KΩ

  P = 100 watts, R_1 = 38KΩ

20

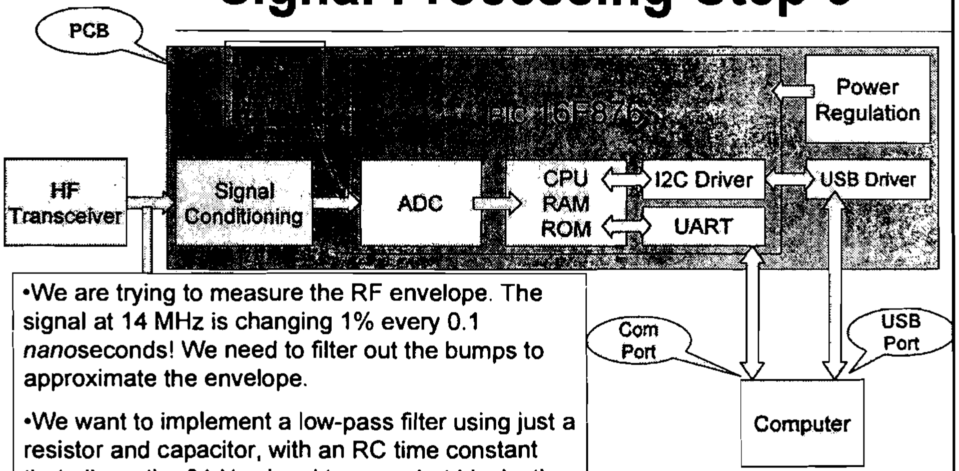


Signal Processing-Step 2

- Signal is AC. ADC sampling will not occur in any definite phase, so we will get an undesirable mix of both positive and negative samples.
- Solution: half-wave rectify the signal prior to sampling using a germanium diode. Why half-wave and not full-wave? Why germanium and not silicon?

21

Signal Processing-Step 3



- We are trying to measure the RF envelope. The signal at 14 MHz is changing 1% every 0.1 *nano*seconds! We need to filter out the bumps to approximate the envelope.
- We want to implement a low-pass filter using just a resistor and capacitor, with an RC time constant that allows the 31 Hz signal to pass, but blocks the RF.
- The PIC's ADC has an internal input capacitance of 5 pf. So the question is: how to determine the value of the resistor?

Signal Processing-Step 3

The cutoff frequency in this low pass filter is

$f_c = 1/(2 \pi RC)$

And the "gain" of the filter is given by

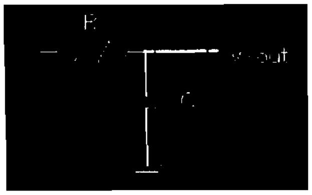
$G = 1/(1 + (f/f_c)^2)^{1/2}$

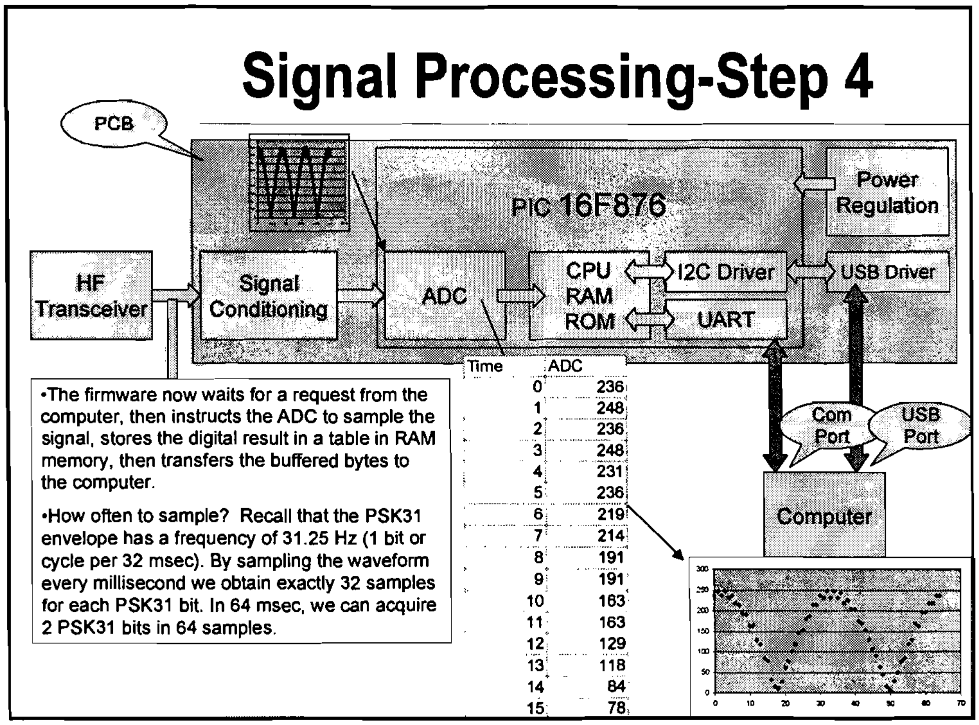
A good cutoff to choose would be the geometric mean between 31 Hz (the signaling envelope) and 14 MHz (the carrier frequency). This works out to be 21 KHz.

Then G for the RF component is 1.5×10^{-3} and for the PSK31 component is $1 - 10^{-6}$.

Since C=5pf, the value of R works out to be

$R = 1/(2 \pi f_c C) = 1.5M\Omega$



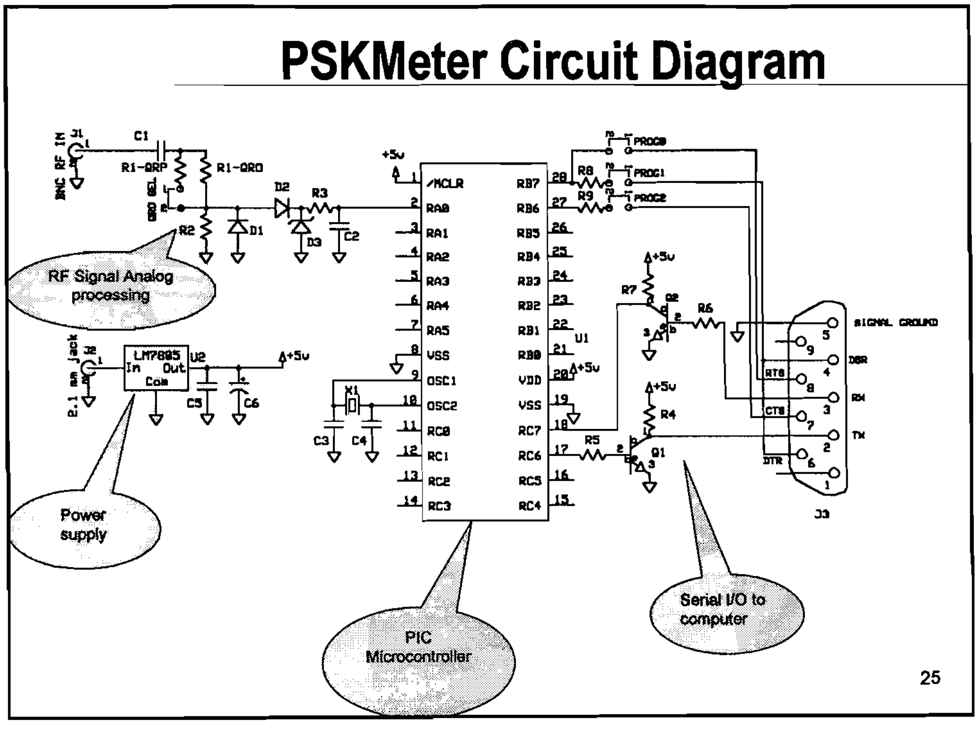


Signal Processing-Step 4

•The firmware now waits for a request from the computer, then instructs the ADC to sample the signal, stores the digital result in a table in RAM memory, then transfers the buffered bytes to the computer.

•How often to sample? Recall that the PSK31 envelope has a frequency of 31.25 Hz (1 bit or cycle per 32 msec). By sampling the waveform every millisecond we obtain exactly 32 samples for each PSK31 bit. In 64 msec, we can acquire 2 PSK31 bits in 64 samples.

Time	ADC
0	236
1	248
2	236
3	248
4	231
5	236
6	219
7	214
8	191
9	191
10	163
11	163
12	129
13	118
14	84
15	78



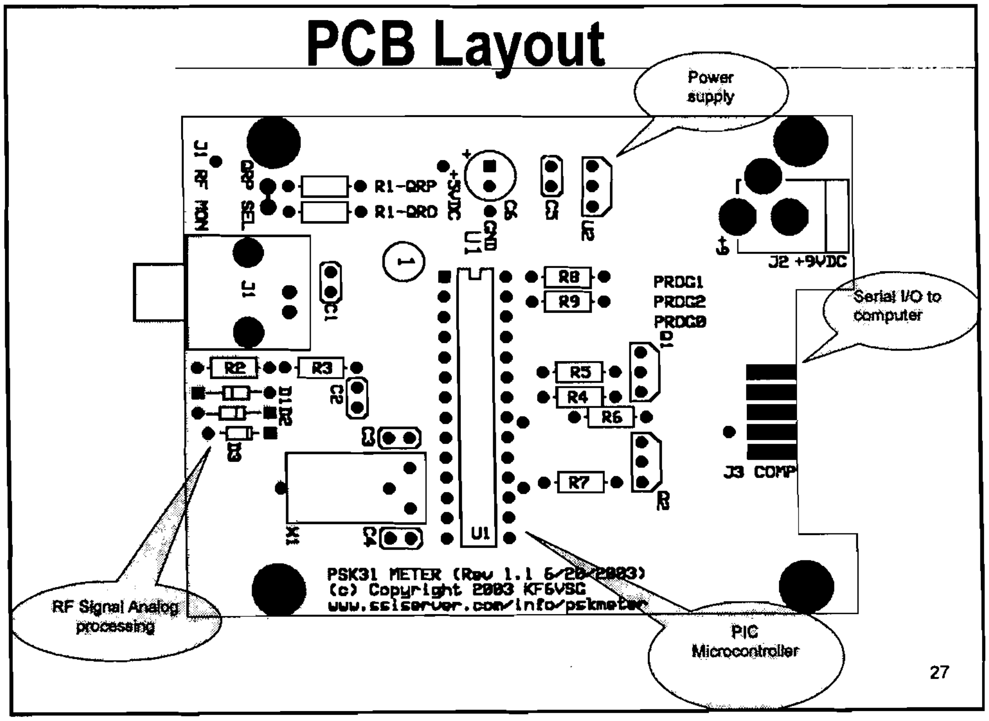
PSKMeter Circuit Diagram

PCB Layout

- In theory the circuit should and does work, but…
- You have RF signals and digital signals on the same board…so what?
- Strong RF signals at 14 MHz will mix with the 20MHz crystal and shift the CPU's clock, and the firmware will crash
- RF should be isolated from any signals that could creep into your computer, or you will have software problems
- The solution is to *be careful with the physical layout* of the board:
  - Keep all RF traces as short as possible.
  - Use plenty of ground plane on the copper side of the board, and even the component side also.
  - Keep the PIC and its crystal away from the RF signal conditioning components and traces.
  - Use ferrite beads (toroids) on the cables to your computer.

26

PCB Layout



Power supply

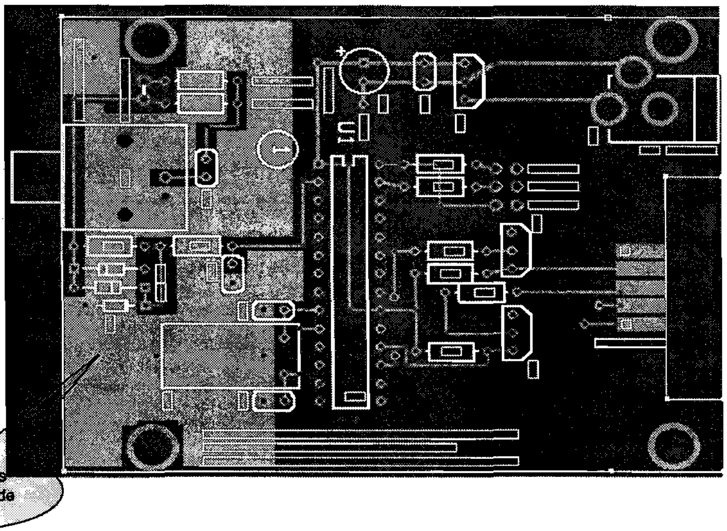
Serial I/O to computer

RF Signal Analog processing

PIC Microcontroller

27

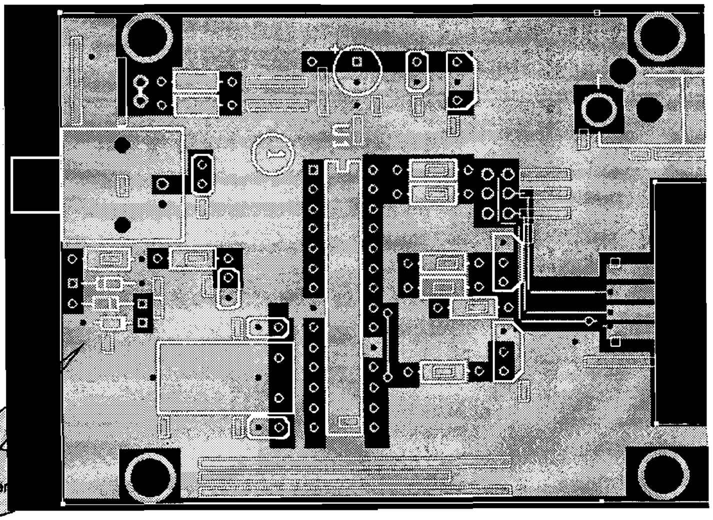
Component Side Ground Plane



Red indicates component side copper

28

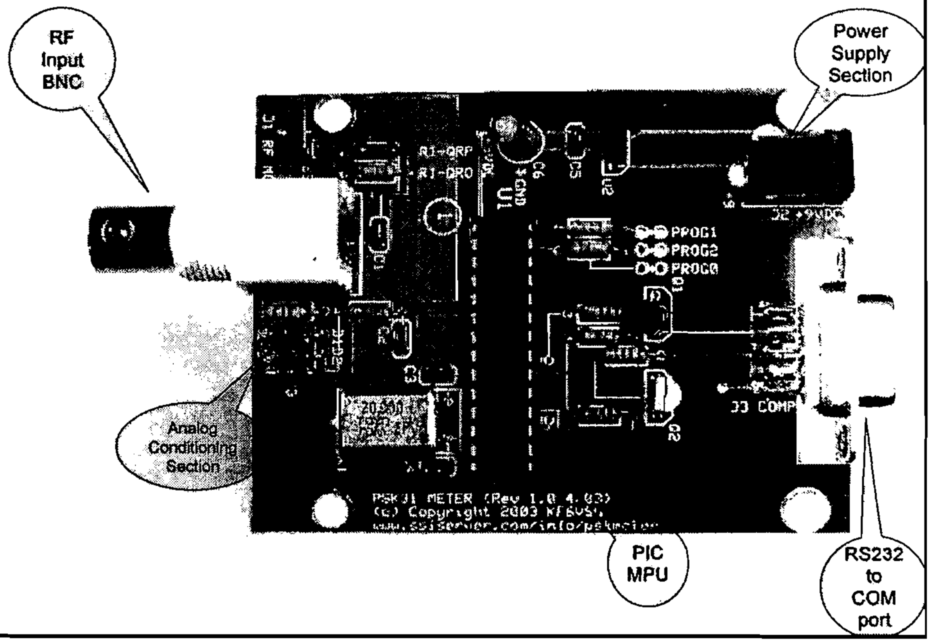
Solder Side Ground Plane



Green indicates solder side coppe

29

PSK Meter Printed Circuit Board



RF Input BNC

Power Supply Section

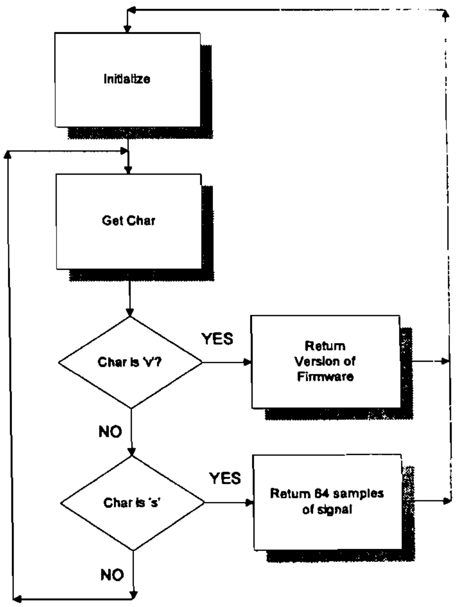
Analog Conditioning Section

PIC MPU

RS232 to COM port

Firmware

- Firmware: software that runs on the PIC in ROM
- Initializes hardware and variables
- Waits for a character command from the PC
- Returns either
  - information about itself, or
  - takes 64 samples of the RF envelope at 1 msec intervals



Initialize

Get Char

Char is 'V'?

YES

Return Version of Firmware

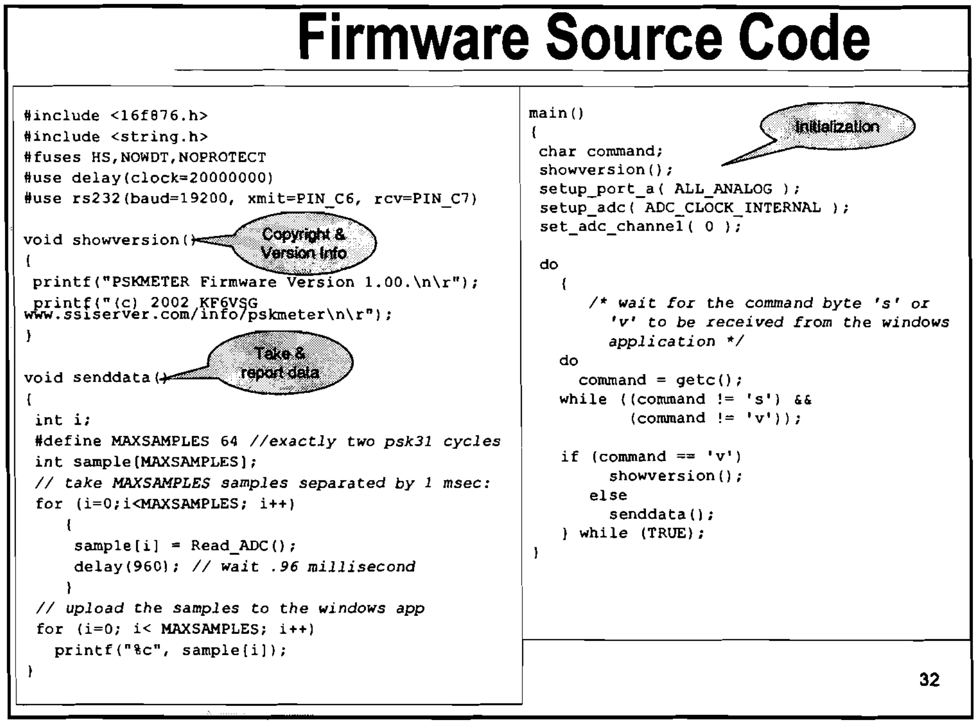
NO

Char is 's'

YES

Return 64 samples of signal

NO



Firmware Source Code

```
#include <16f876.h>
#include <string.h>
#fuses HS,NOWDT,NOPROTECT
#use delay(clock=20000000)
#use rs232(baud=19200, xmit=PIN_C6, rcv=PIN_C7)

void showversion()
{
 printf("PSKMETER Firmware Version 1.00.\n\r");
 printf("(c) 2002 KF6VSG
www.ssiserver.com/info/pskmeter\n\r");
}

void senddata()
{
 int i;
 #define MAXSAMPLES 64 //exactly two psk31 cycles
 int sample[MAXSAMPLES];
 // take MAXSAMPLES samples separated by 1 msec:
 for (i=0;i<MAXSAMPLES; i++)
    {
     sample[i] = Read_ADC();
     delay(960); // wait .96 millisecond
    }
 // upload the samples to the windows app
 for (i=0; i< MAXSAMPLES; i++)
   printf("%c", sample[i]);
}
```

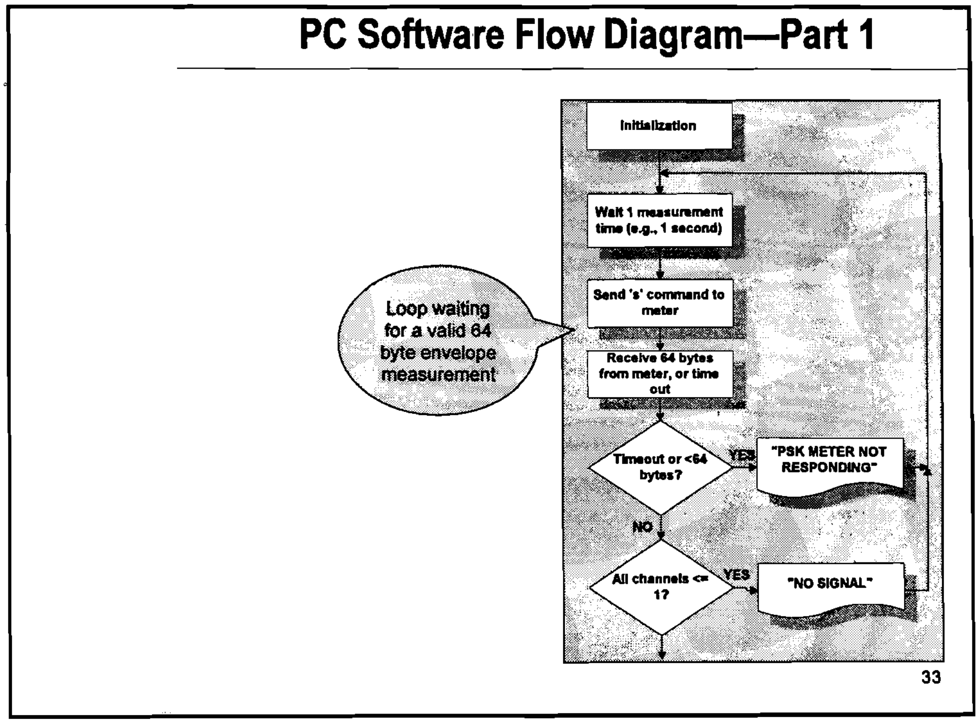
```
main()
{
 char command;
 showversion();
 setup_port_a( ALL_ANALOG );
 setup_adc( ADC_CLOCK_INTERNAL );
 set_adc_channel( 0 );

 do
   {
     /* wait for the command byte 's' or
        'v' to be received from the windows
        application */
   do
     command = getc();
   while ((command != 's') &&
          (command != 'v'));

   if (command == 'v')
        showversion();
     else
        senddata();
   } while (TRUE);
}
```

Copyright & Version Info

Take & report data

Initialization

32

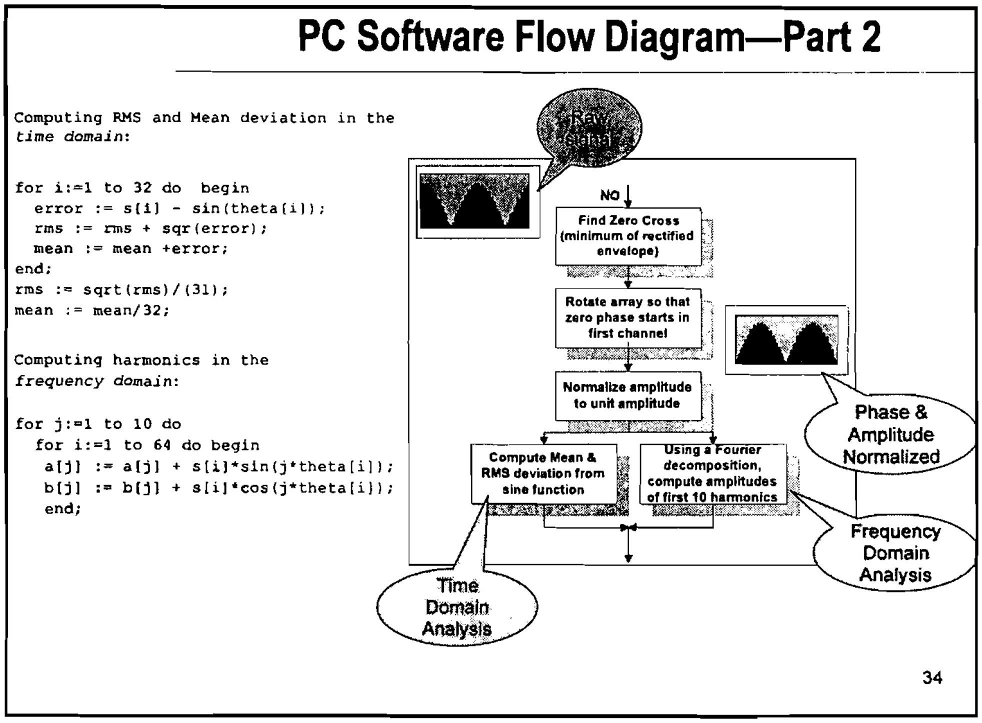


PC Software Flow Diagram—Part 1

Loop waiting for a valid 64 byte envelope measurement

Initialization
Wait 1 measurement time (e.g., 1 second)
Send 's' command to meter
Receive 64 bytes from meter, or time out
Timeout or <64 bytes?
YES
"PSK METER NOT RESPONDING"
NO
All channels <= 1?
YES
"NO SIGNAL"

33



PC Software Flow Diagram—Part 2

Computing RMS and Mean deviation in the *time domain*:

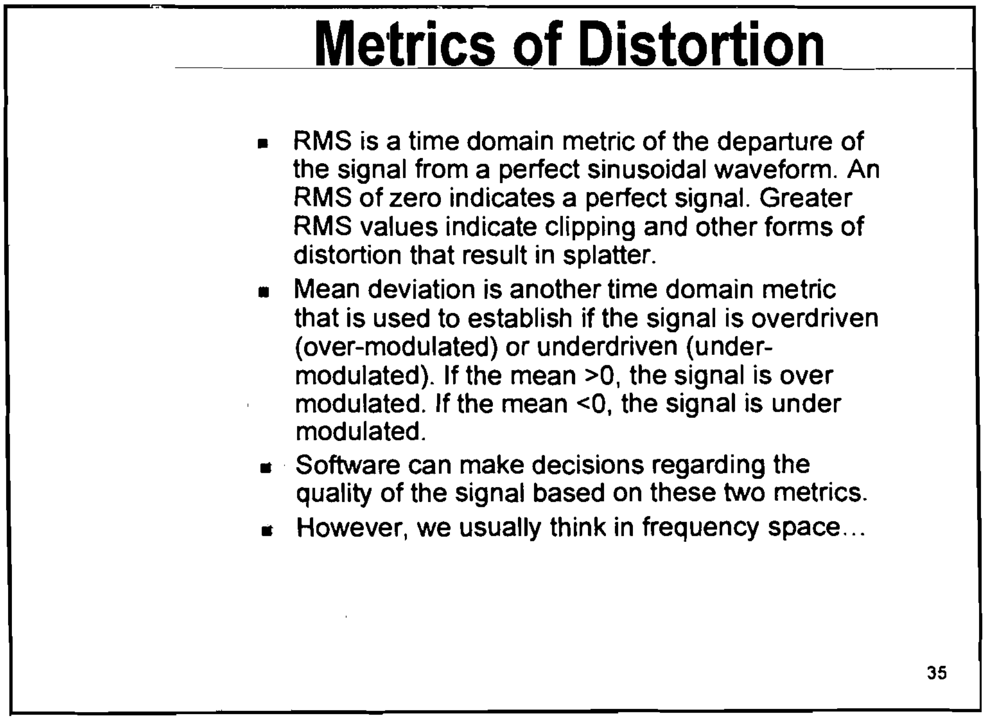
```
for i:=1 to 32 do  begin
  error := s[i] - sin(theta[i]);
  rms := rms + sqr(error);
  mean := mean +error;
end;
rms := sqrt(rms)/(31);
mean := mean/32;
```

Computing harmonics in the *frequency domain*:

```
for j:=1 to 10 do
  for i:=1 to 64 do begin
   a[j] := a[j] + s[i]*sin(j*theta[i]);
   b[j] := b[j] + s[i]*cos(j*theta[i]);
   end;
```

Raw Signal
NO
Find Zero Cross (minimum of rectified envelope)
Rotate array so that zero phase starts in first channel
Normalize amplitude to unit amplitude
Compute Mean & RMS deviation from sine function
Using a Fourier decomposition, compute amplitudes of first 10 harmonics
Phase & Amplitude Normalized
Frequency Domain Analysis
Time Domain Analysis

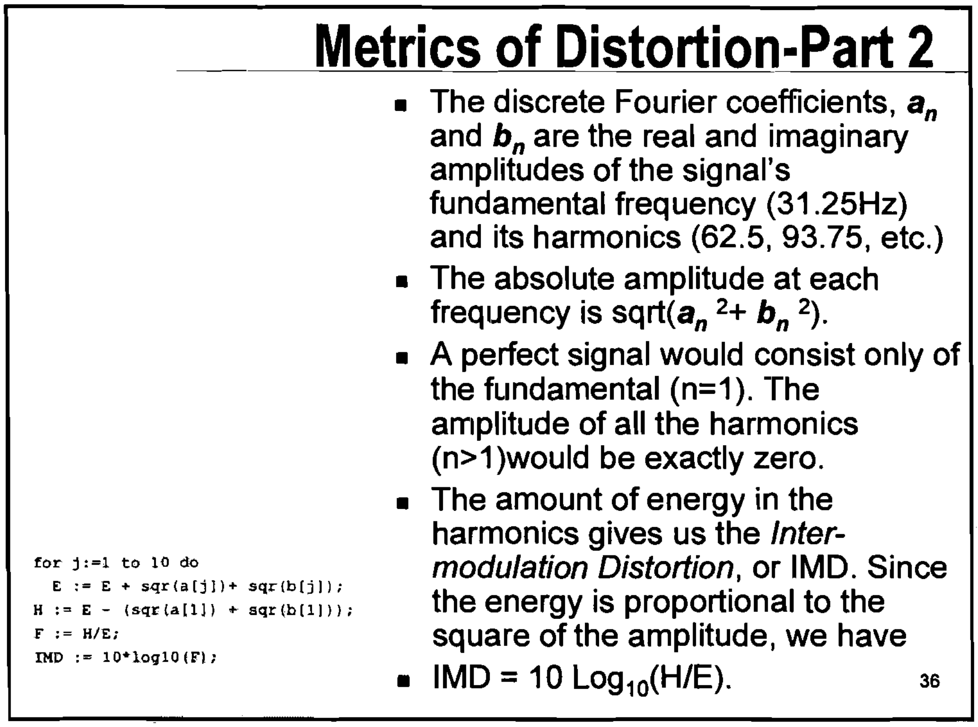
34



Metrics of Distortion

- RMS is a time domain metric of the departure of the signal from a perfect sinusoidal waveform. An RMS of zero indicates a perfect signal. Greater RMS values indicate clipping and other forms of distortion that result in splatter.
- Mean deviation is another time domain metric that is used to establish if the signal is overdriven (over-modulated) or underdriven (under-modulated). If the mean >0, the signal is over modulated. If the mean <0, the signal is under modulated.
- Software can make decisions regarding the quality of the signal based on these two metrics.
- However, we usually think in frequency space...

35

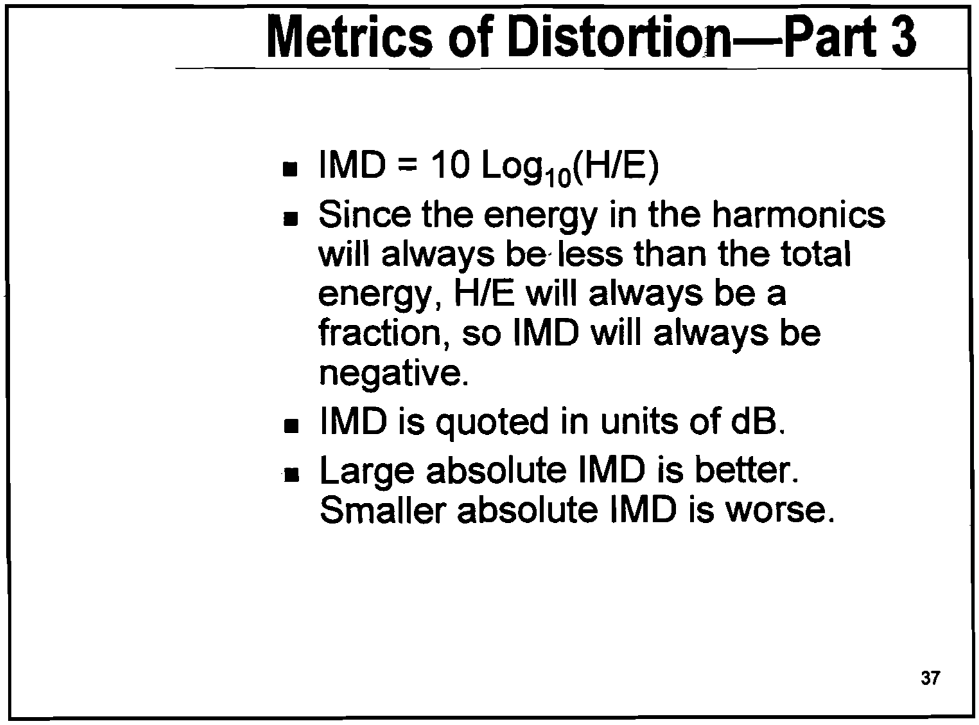


Metrics of Distortion-Part 2

- The discrete Fourier coefficients, $\boldsymbol{a_n}$ and $\boldsymbol{b_n}$ are the real and imaginary amplitudes of the signal's fundamental frequency (31.25Hz) and its harmonics (62.5, 93.75, etc.)
- The absolute amplitude at each frequency is sqrt($\boldsymbol{a_n}$ ^{2}+ $\boldsymbol{b_n}$ ^{2}).
- A perfect signal would consist only of the fundamental (n=1). The amplitude of all the harmonics (n>1)would be exactly zero.
- The amount of energy in the harmonics gives us the *Inter-modulation Distortion*, or IMD. Since the energy is proportional to the square of the amplitude, we have
- IMD = 10 Log_{10}(H/E).

```
for j:=1 to 10 do
  E := E + sqr(a[j])+ sqr(b[j]);
H := E - (sqr(a[1]) + sqr(b[1]));
F := H/E;
IMD := 10*log10(F);
```

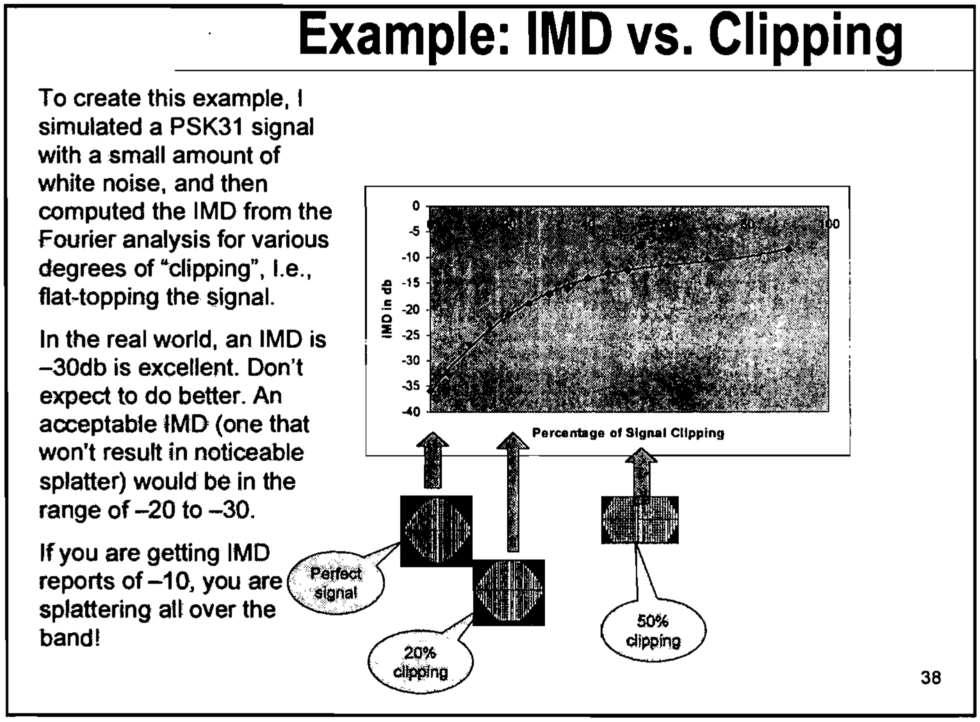
36



Metrics of Distortion—Part 3

- IMD = 10 Log_{10}(H/E)
- Since the energy in the harmonics will always be less than the total energy, H/E will always be a fraction, so IMD will always be negative.
- IMD is quoted in units of dB.
- Large absolute IMD is better. Smaller absolute IMD is worse.

37



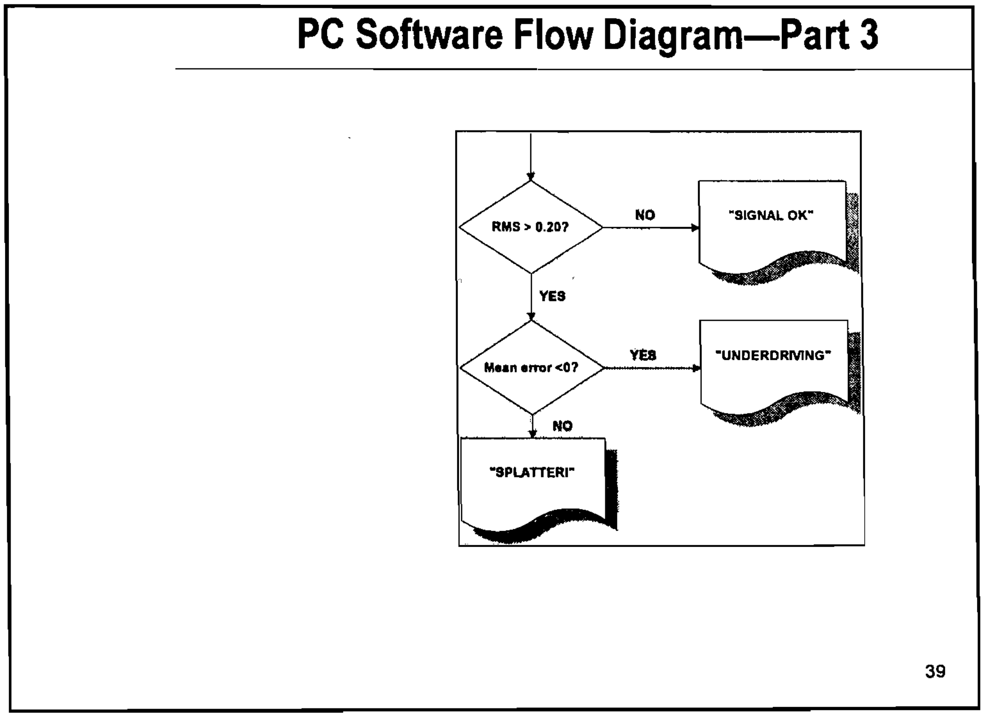
Example: IMD vs. Clipping

To create this example, I simulated a PSK31 signal with a small amount of white noise, and then computed the IMD from the Fourier analysis for various degrees of "clipping", I.e., flat-topping the signal.

In the real world, an IMD is –30db is excellent. Don't expect to do better. An acceptable IMD (one that won't result in noticeable splatter) would be in the range of –20 to –30.

If you are getting IMD reports of –10, you are splattering all over the band!

38



PC Software Flow Diagram—Part 3

RMS > 0.20?

NO

"SIGNAL OK"

YES

Mean error <0?

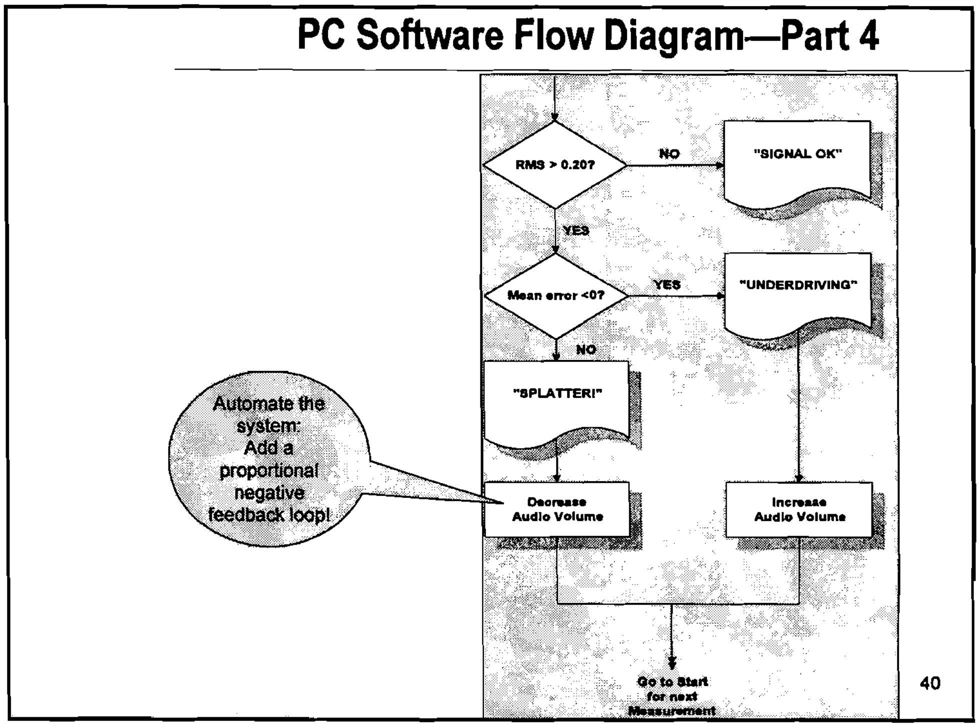
YES

"UNDERDRIVING"

NO

"SPLATTERI"

39



PC Software Flow Diagram—Part 4

RMS > 0.20?

NO

"SIGNAL OK"

YES

Mean error <0?

YES

"UNDERDRIVING"

NO

"SPLATTER!"

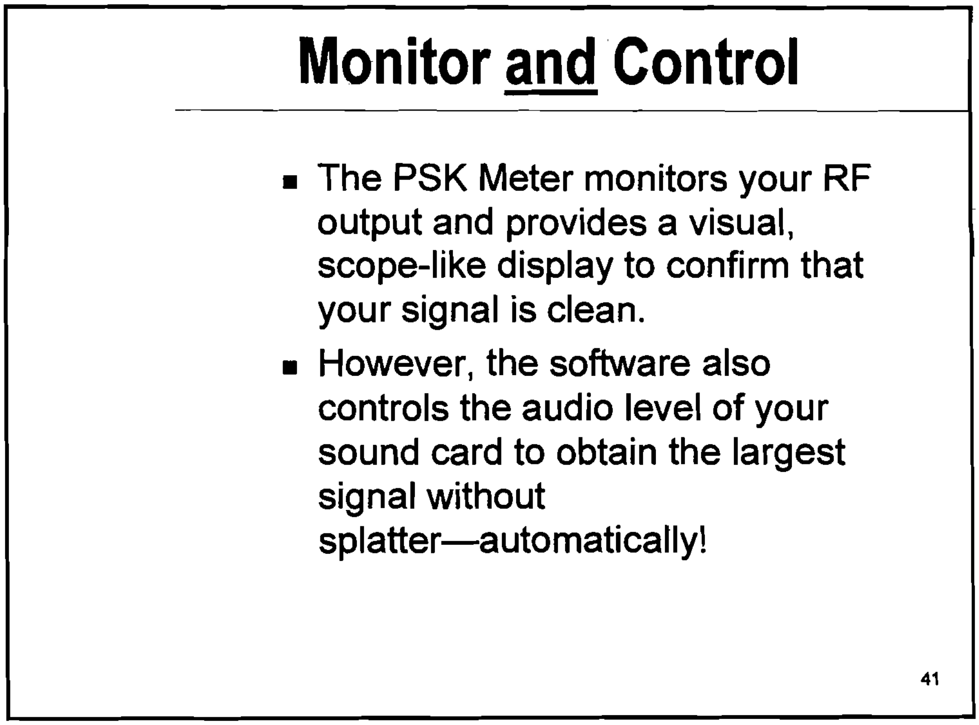
Decrease Audio Volume

Increase Audio Volume

Go to Start for next Measurement

Automate the system: Add a proportional negative feedback loop!

40



Monitor and Control

- The PSK Meter monitors your RF output and provides a visual, scope-like display to confirm that your signal is clean.
- However, the software also controls the audio level of your sound card to obtain the largest signal without splatter—automatically!

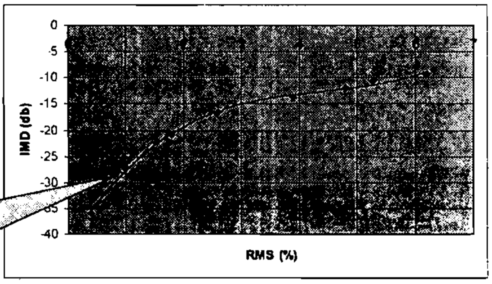
41

Control variables—what to use?

•RMS (deviation from ideal signal), in percent

•IMD (harmonic disortion), in dB

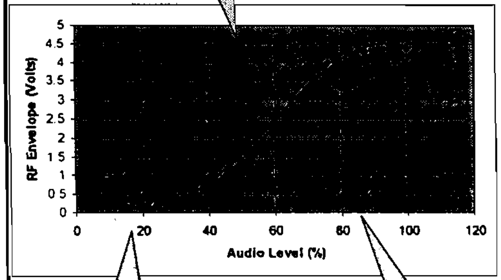
•Average deviation—discerns between overmodulation and undermodulation



In most cases, RMS and IMD can be used interchangably as the control variable

A graphical look at feedback

Region of Optimum Audio Level



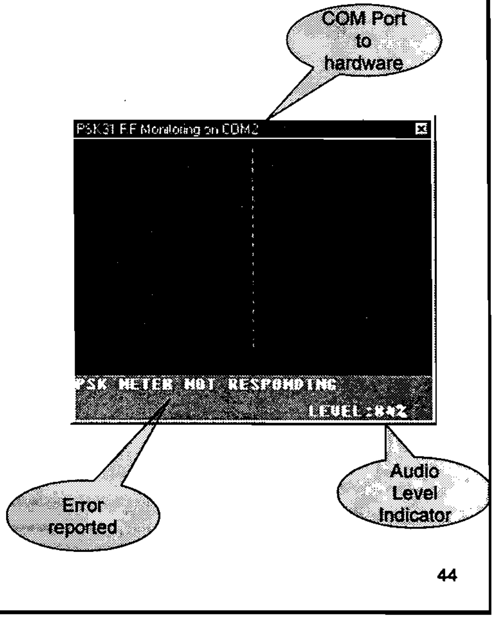
Undermodulation Region

Distortion Region

- Plot the peak PSK31 envelope vs audio level.
- Notice that clipping occurs for the top 1/3 in audio level.
- Define underdriving as the bottom 1/3 in audio level.
- The feedback loop in the software attempts to find and place the audio level in the middle third, tending to the largest signal without distortion.

A look at the PSKMeter Software

- When PSKMeter is launched, it displays the COM port it uses to connect to the PSKMeter hardware.
- If the hardware fails to respond, a message is displayed.
  - Wrong COM Port
  - No serial cable
  - No PSKMeter hardware
  - PSKMeter hardware not powered up



PSK31 RF Monitoring on COM2

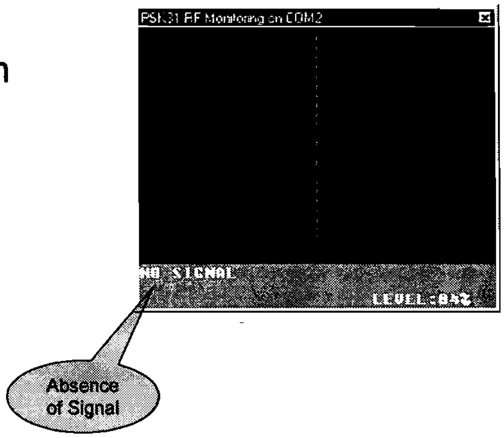
PSK METER NOT RESPONDING

LEVEL:84%

44

A look at the PSKMeter Software

- If the hardware connects, the software periodically issues a request for waveform sample.
- When you are not in transmit mode, no signal is returned.



PSK31 RF Monitoring on COM2

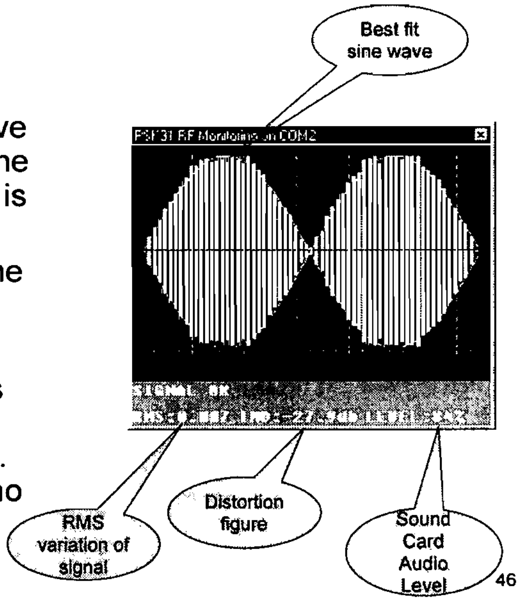
NO SIGNAL

LEVEL:84%

45

A look at the PSKMeter Software

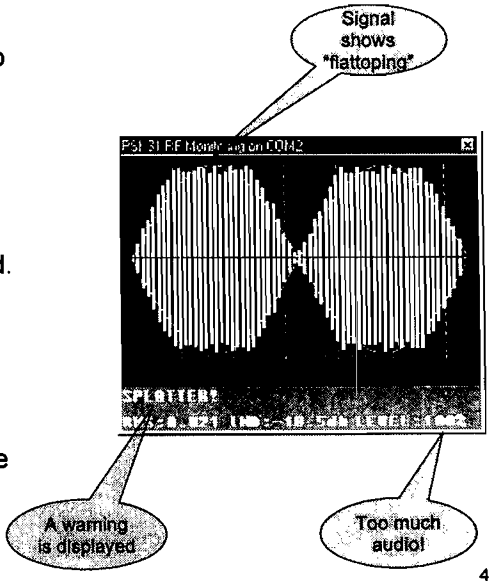
- In transmit mode, the RF output is periodically sampled and displayed.
- The best-fit sine wave curve is displayed. Variation of the signal from this sine wave is a measure of distortion.
- RMS variation from the sine wave is displayed.
- IMD is displayed.
- If RMS is small and IMD is large, the indication "SIGNAL OK" is displayed. Your signal is clean, and no adjustment (automatic or manual) is necessary.



46

A look at the PSKMeter Software

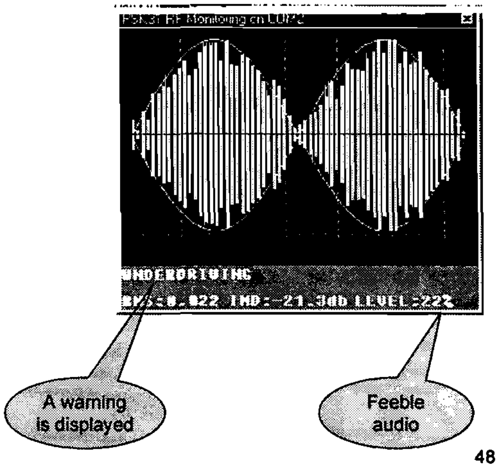
- If you are over-driving your transmitter, your signal is going to show visible evidence of distortion.
- Example: flat-topping (clipping).
- The RMS will be large and the IMD will be small.
- The result is splatter, as indicated. You are causing interference to other portions of the band, and your signal will not be as copiable!
- When in automatic mode, the software will automatically reduce the sound card's audio level to completely remove the splatter.



47

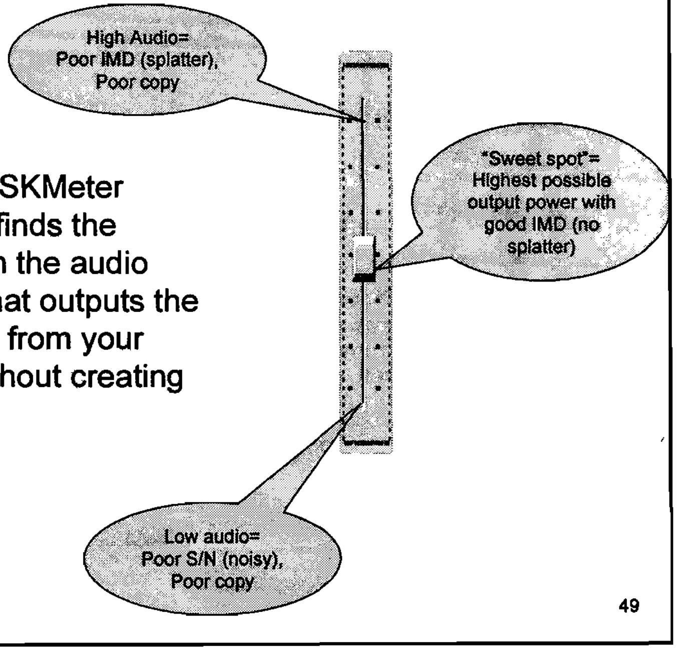
A look at the PSKMeter Software

- If you are under-driving your transmitter, your signal will be weak and noisy, but not distorted.
- Because your signal is weak, you will not be as copiable.
- When in automatic mode, the software will automatically increase the sound card's audio level, boosting your output power without introducing distortion.



48

PSKMeter Benefit:

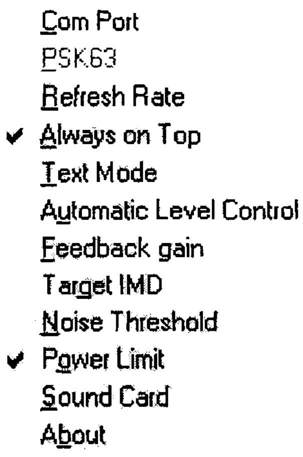

High Audio= Poor IMD (splatter), Poor copy
"Sweet spot"= Highest possible output power with good IMD (no splatter)
Low audio= Poor S/N (noisy), Poor copy

- Bottom line: PSKMeter *automatically* finds the "sweet spot" in the audio level setting that outputs the highest power from your transmitter without creating splatter.

49

PSKMeter User Interface

- Right-click on the meter brings up the user interface menu:
- Com port selection
- Mode selection (PSK31/PSK63)
- Refresh (sampling) rate
- Force window to be always visible
- View data numerically
- Force automatic control of audio level
- Gain selection
- Desired IMD selection
- Noise (floor) threshold
- Maximum power emission selection
- Sound card selection



50

Mode Selection

- “Standard” mode is PSK31 (31.25 Hz = 32 msec per bit)
- Recently we have the ability to transmit PSK63 (62.50 Hz = 16 msec per bit)
- By selecting PSK63, a command it sent to the firmware that instructs it to use ½ msec sampling time instead of 1 msec. Everything else works the same way.

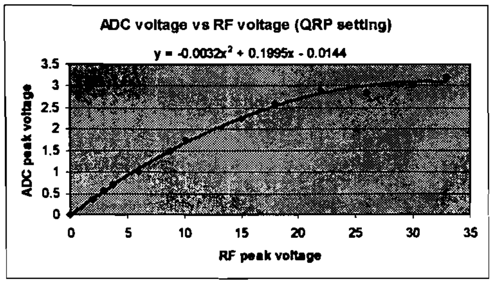
51

Maximum Power Selection

- PSKMeter is capable of controlling the power output to a specified level
- From $P=E^2/2R = E^2/100$, we know the peak RF voltage given an average power level (i.e., E=10 sqrt(P)).
- But what is the relationship between the peak voltage seen at the ADC to the peak RF voltage?

ADC voltage vs RF voltage

- Shown below is a measurement of the signal at the ADC as a function of the RF peak voltage:



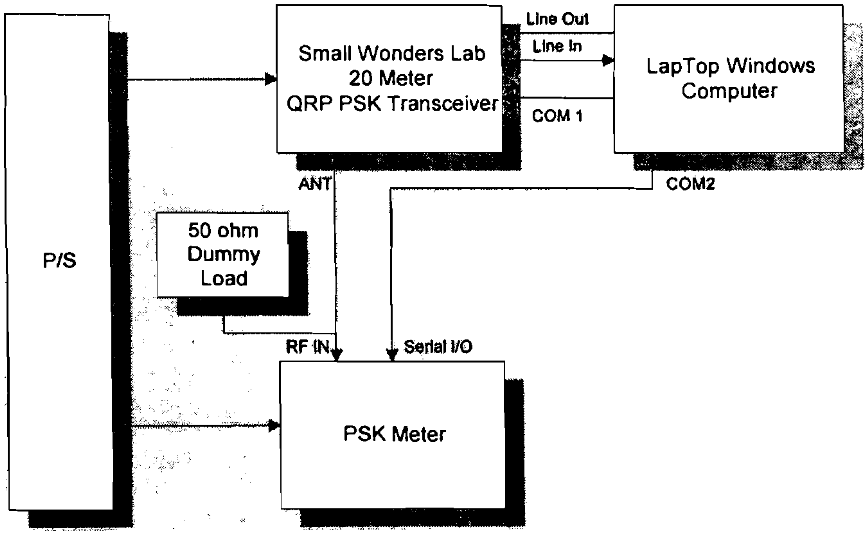
- Notice that the relationship is linear only for low power. The function is well approximated by a cubic polynomial over a wide range of power.

Maximum Power Selection

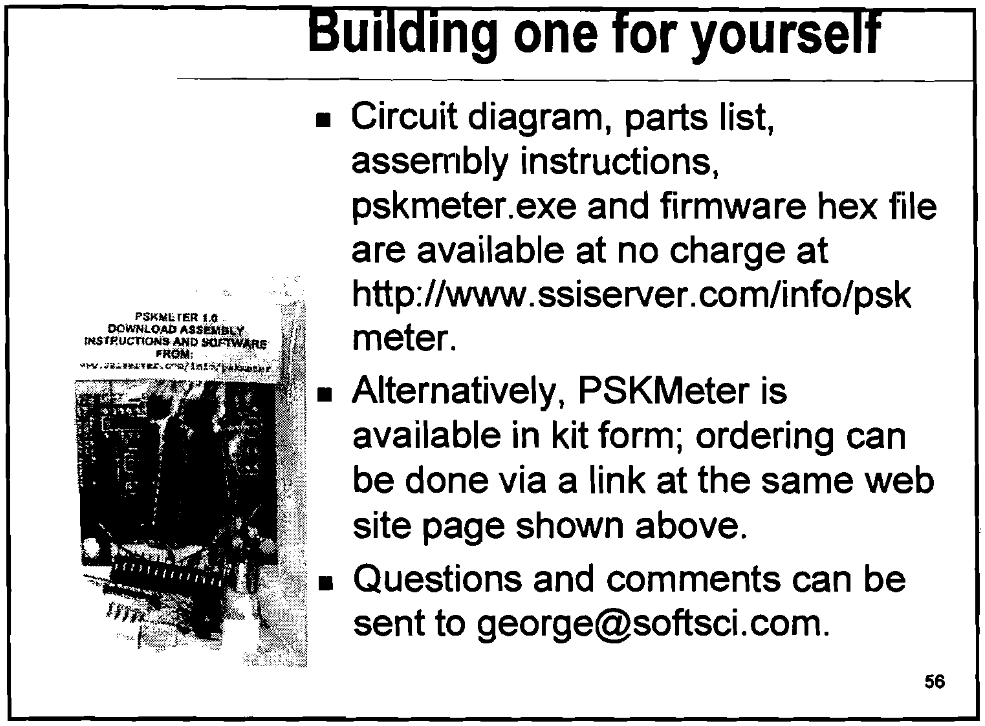
- Thus, given a desired power level, the software computes the RF peak voltage, then using this polynomial fit to the empirical data determines the voltage at the input to the ADC.
- The software then sets the audio level to obtain this voltage, thereby achieving the desired radiated power.

54

Demonstration Time!



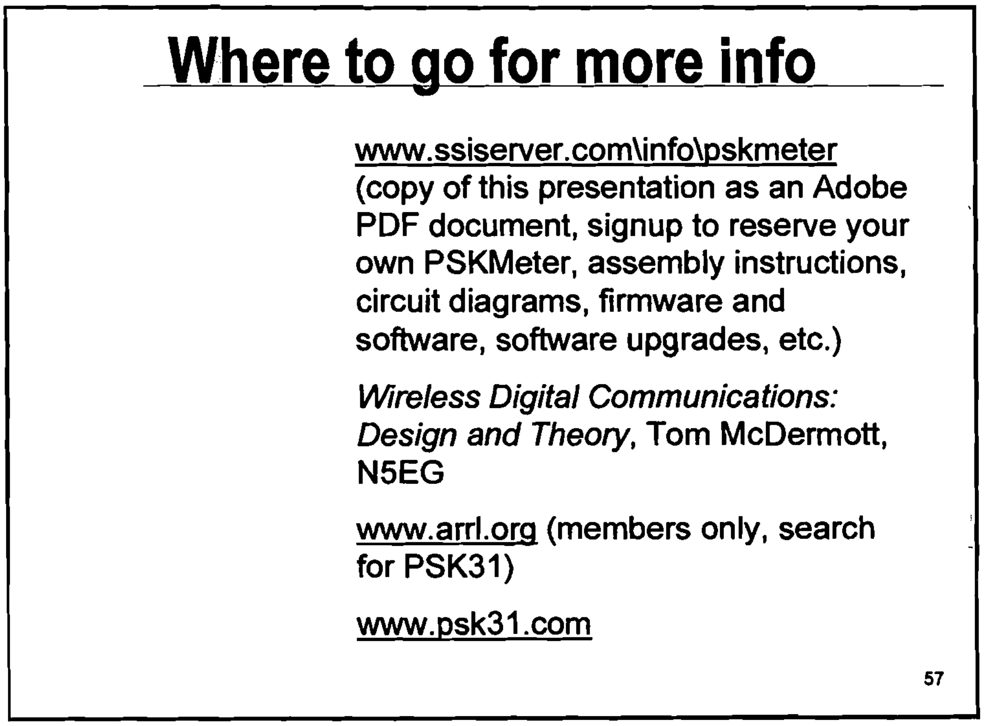
55



Building one for yourself

- Circuit diagram, parts list, assembly instructions, pskmeter.exe and firmware hex file are available at no charge at http://www.ssiserver.com/info/pskmeter.
- Alternatively, PSKMeter is available in kit form; ordering can be done via a link at the same web site page shown above.
- Questions and comments can be sent to george@softsci.com.

56



Where to go for more info

www.ssiserver.com\info\pskmeter (copy of this presentation as an Adobe PDF document, signup to reserve your own PSKMeter, assembly instructions, circuit diagrams, firmware and software, software upgrades, etc.)

Wireless Digital Communications: Design and Theory, Tom McDermott, N5EG

www.arrl.org (members only, search for PSK31)

www.psk31.com

57