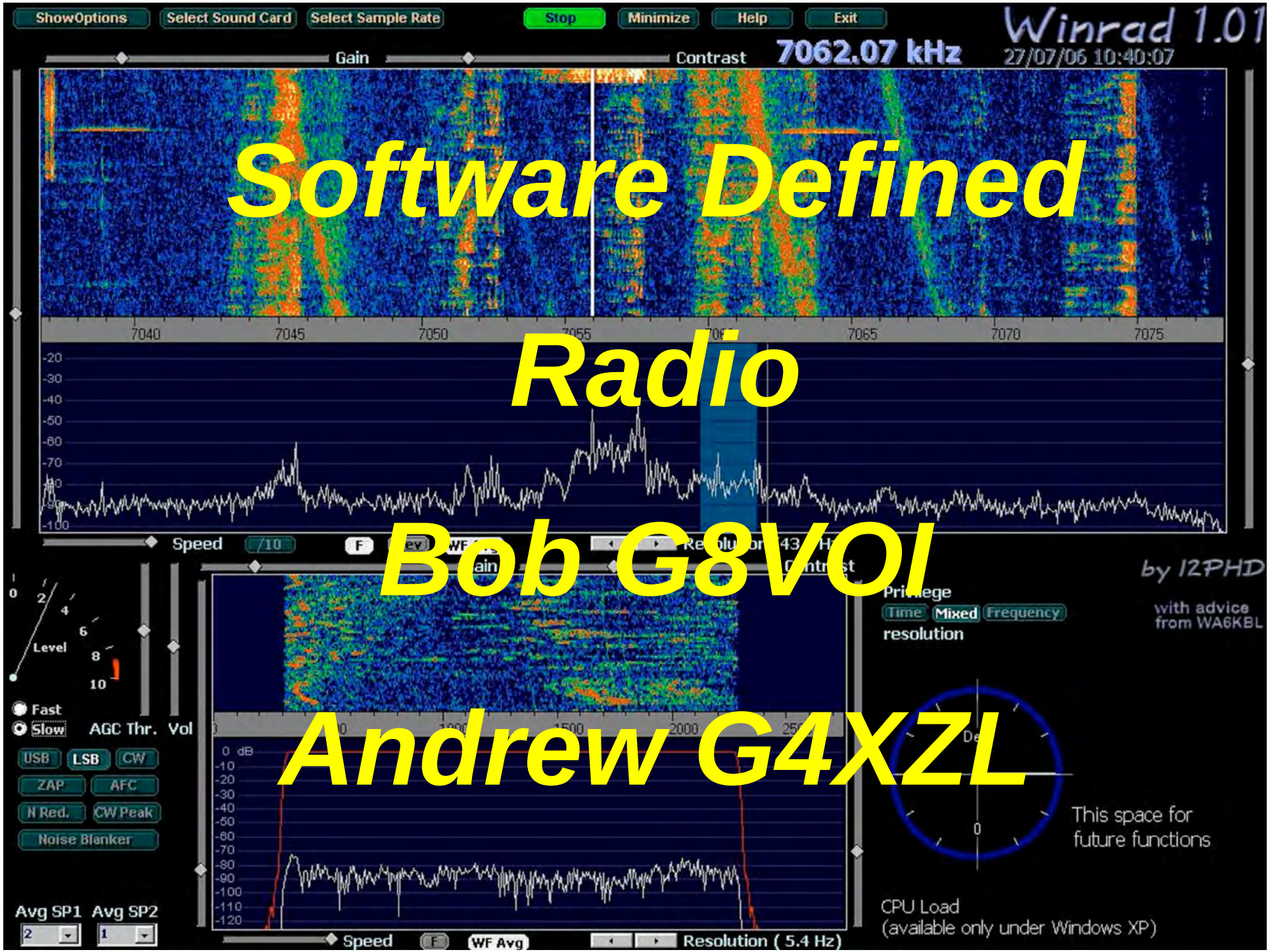

ShowOptions
Select Sound Card
Select Sample Rate
Stop
Minimize
Help
Exit
Winrad 1.01
Gain
Contrast
7062.07 kHz
27/07/06 10:40:07
Software Defined
Radio
Bob G8VOI
Andrew G4XZL
Speed
Level
Fast
Slow
AGC Thr.
Vol
USB
LSB
CW
ZAP
AFC
N Red.
CW Peak
Noise Blanker
Avg SP1
Avg SP2
Speed
WF Avg
Resolution (5.4 Hz)
by I2PHD
with advice
from WA6KBL
Time
Mixed
Frequency
resolution
This space for
future functions
CPU Load
(available only under Windows XP)

Introduction

In the last year or so articles and reviews have started to appear in ***Radcom*** and ***Practical Wireless*** talking about ***Software Defined Radio (SDR)***

The first amateur ***SDR*** transceivers ***(SDR-1000)*** have appeared on the market, and receiver kits can easily be obtained.

This talk is based around the simple ***SoftRock SDR*** receiver kit costing ***< £10!***

The Aim for Tonight?

i) Provide a basic introduction to ***SDR***

ii) Look at the ***SoftRock*** receiver kits

iii) Demonstrate different software

iv) Most importantly, give opportunity try it and make your own mind up!

What is it SDR?

Like it or not, this is probably the way ahead in the future and should hopefully offer cheaper, more versatile transceivers.

The speed of home PC's and the power of the DSP within the soundcard continues to increase, as the cost decreases.

Conventional transceivers are very expensive to produce and limited to the features provided by the initial designers.

You have two choices:

1) The ‘luddites’ dismiss ***SDR*** as rubbish.

2) The ‘curious’ try to find out more!

There can be some confusion :

SDR is ***NOT*** using a conventional ‘black box’ radio connected to a PC via an interface such as RS232, C-IV, CAT or USB.

This is merely ***controlling*** the radio via computer.

The main features of ***SDR*** is that the PC carries out all of the demodulation (or modulation in a TX) of the RF signal.

This is done by using an external box to perform the basic down conversion of the incoming RF signal to a low I.F. (e.g. 11kHz) or base-band frequency block.

This can be fed directly to the audio line inputs of the PC soundcard for processing.

All ***'features'*** of the 'radio' are ***defined*** by the operating software running on the PC, therefore the same external converter can be used with many different software packages tailored to suit individual requirements.

A number of ***SDR*** projects have previously been published in journals such as ***QST*** and ***Dubus.***

These used external DSP evaluation boards connected to a PC, an example is the ***Leif SM5BSZ Linux*** based ***Linrad*** system.

Within the past year, PC soundcard based receiver kits ***(SoftRock)*** and complete transceivers ***(SDR-1000)*** have appeared.

In the future there will be no need to replace your transceiver to upgrade it, it will merely require a download of a new software package to acquire the latest features.

It should be possible to incorporate all ‘data’ modes within the package, making it a truly all mode transceiver.

At the present time, all ***SDR*** products should be viewed as ***'cutting edge' or experimental***.

There are short comings in some aspects of the performance, namely image rejection and strong signal handling, however it is very early days with this technology and things are bound to improve rapidly.

Look at what has been achieved rather than just focussing on the weaker points.

It is too easy to become paranoid with performance figures, and loose sight of what the ***SoftRock*** receivers are trying to achieve.

It is a cheap, simple means of enabling anyone to take their first steps in exploring the possibilities of ***SDR*** and furthering their knowledge through experimentation.

Keep an open mind, stop and think about what it can do, then make your own mind up.

The designers of the ***SoftRock SDR*** receiver kits are currently working on a simple, low cost QRP CW / PSK31 ***SDR*** transmitter.

This will create an extremely cheap, but sophisticated QRP ***SDR*** station.

The ***SDR 1000*** first appeared as a ‘homebrew’ magazine project before being offered now as a completed item.

There are currently a number of projects being developed, keep an eye on the internet.

The ***HPSDR*** (***H***igh ***P***erformance ***SDR***) project moves far beyond the simple soundcard systems and uses powerful external dedicated components (brief details in ***September 2006 Radcom***) and on the website ***hpsdr.org***. This is currently being developed by a number of amateurs.

Andy, G4JNT has published in the ***Microwave Newsletter (now Scatterpoint)*** a design for a 144MHz ***SDR*** using a PC controlled DDS synthersizer for the LO.

Chris Bartram, GW4DGU (of ***muTek*** fame) has developed a 1296MHz ***SDR*** and published a number of short articles. The performance is good enough for EME work!

A website dedicated to the use of ***SDR*** for the microwave bands has been set up by ***ON/G4KLX*** to pool hardware and software ideas and designs.

As said before, it is very early days in amateurs using ***SDR*** and progress is very rapid.

What about the 'big' Japanese producers?

Must be looking at ***SDR***, but could be a problem for them.

They would have to produce operating software to go with their hardware.

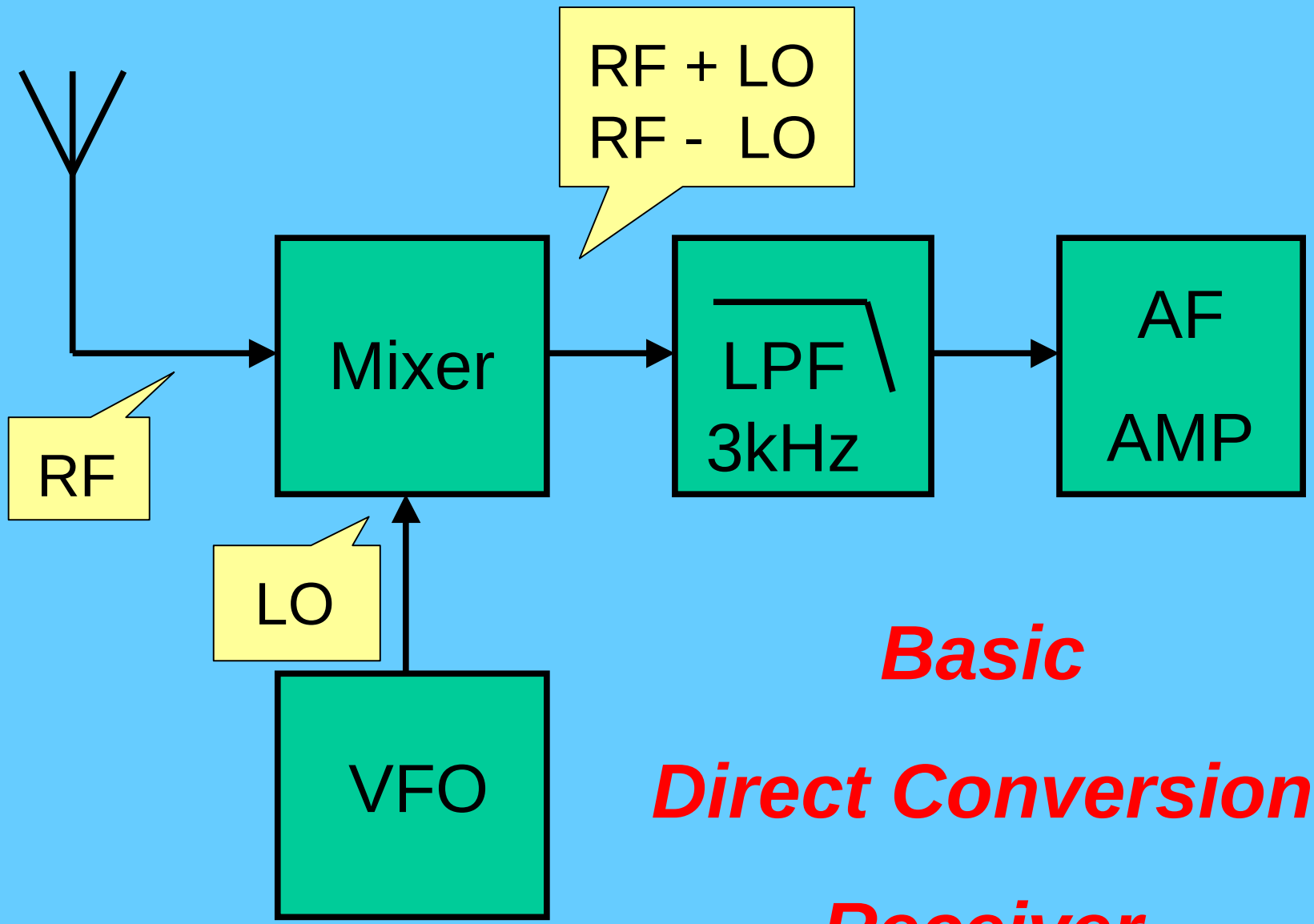
Very unlikely they would make the source code freely available, so it would not fit in with the current way amateur ***SDR*** is developing.

SDR Basics

Probably the first HF receiver most of us built was a simple ***Direct Conversion*** type.

These receivers give good results, but can suffer from a lack of image suppression.

The basic ***SDR*** operates on the same principles as the ***D-C*** receiver, however there are a number of significant differences that lead to the vastly improved performance.



Basic

Direct Conversion

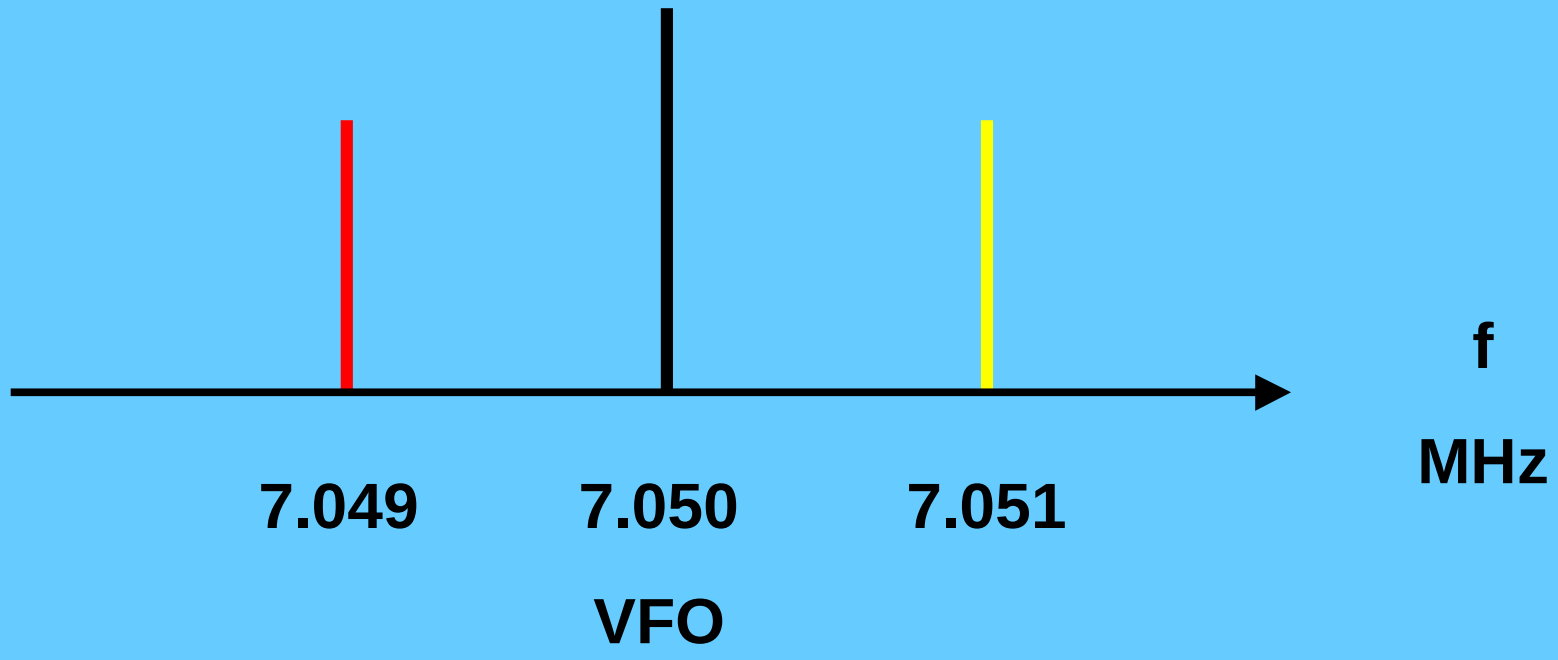
Receiver

The simple ***D-C*** receiver operates by mixing the incoming RF signal with the VFO to produce both sum and difference signals.

The unwanted sum of the two is removed by a Low Pass Filter, leaving only the wanted difference signal (audio frequency).

All incoming signals mix to produce an output, those within the audible range will be heard.

Typically the low pass filter cut off frequency might be set to 3kHz for SSB use, and an additional narrow band-pass filter switched in for CW signals.



Example:

VFO set to: 7.050 MHz

Wanted input: 7.051 MHz

Unwanted input: 7.049 MHz

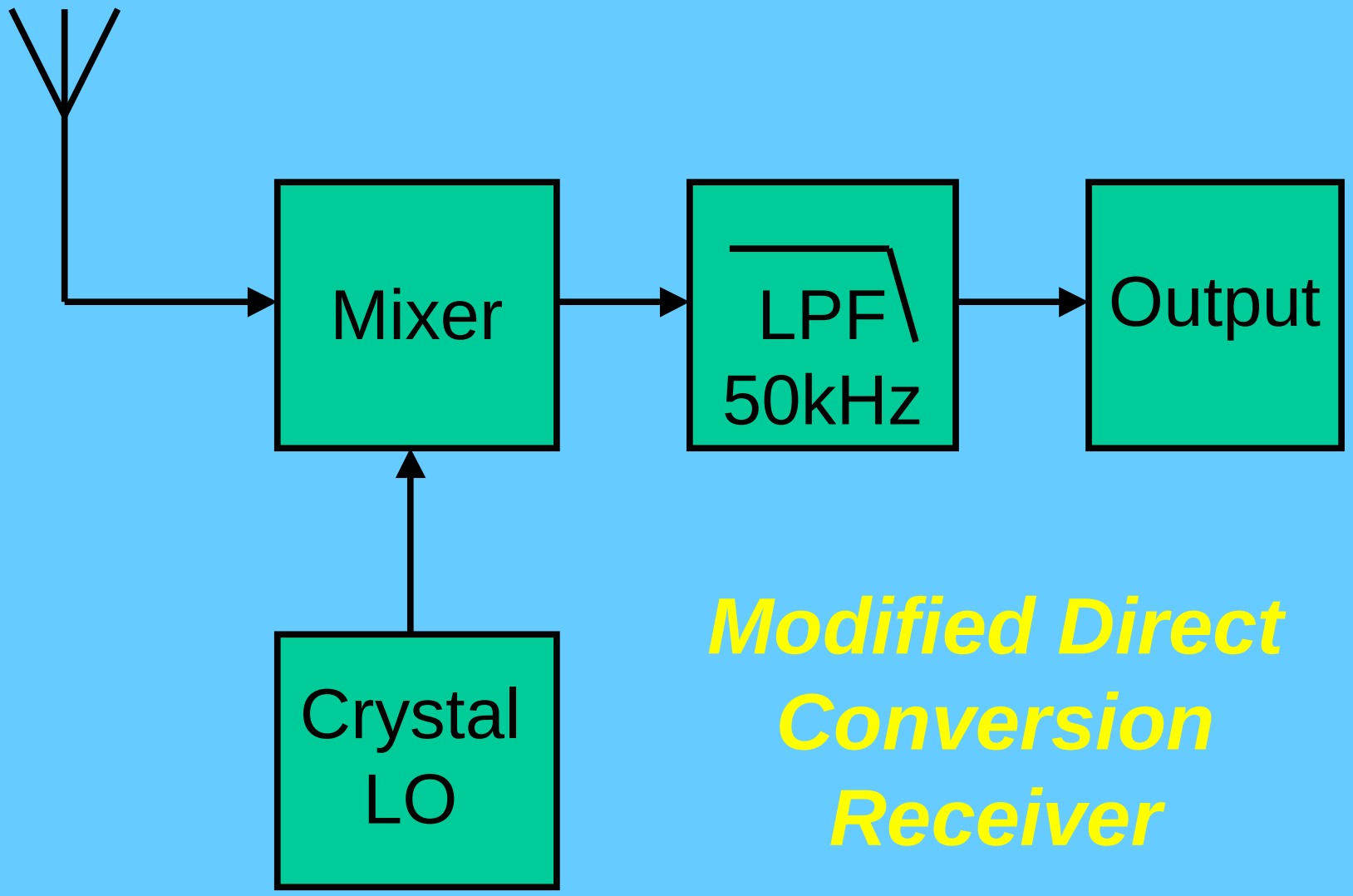
7.051 - 7.050 = 1kHz

7.050 - 7.049 = 1kHz

It can be seen that both input signals produce an output at 1kHz, the ‘classic’ image problem.

Whilst this image problem is unwanted in the simple ***D-C*** receiver, it will be seen later that this can actually be put to good use in the ***SDR***.

The next stage in the transformation from ***D-C*** receiver to simple ***SDR*** is to replace the VFO (Variable Frequency Oscillator) with a fixed frequency source, typically a crystal oscillator.

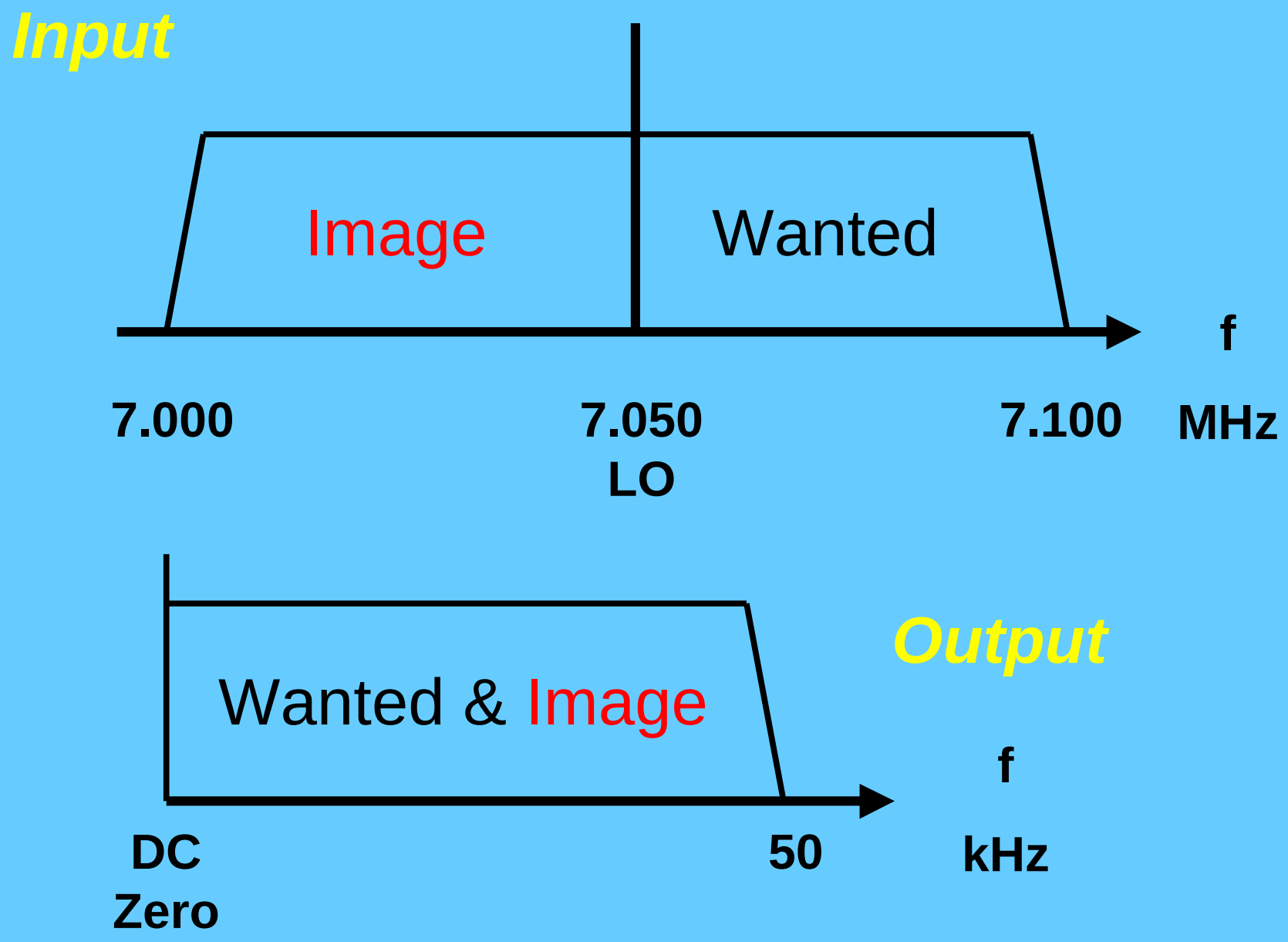


Modified Direct Conversion Receiver

Consider the previous block diagram, the only difference is that the local oscillator is now a fixed source.

The low pass filter after the mixer is modified to have a cut off frequency in the order of 50kHz (above the human hearing range).

All incoming signals that mix with the local oscillator and produce an output in the range DC to 50kHz will be present at the output.


Input
Image
Wanted
f
7.000
7.050
LO
7.100
MHz
Wanted & Image
Output
f
DC
Zero
50
kHz

As can be seen, the output from the modified ***D-C*** receiver is a ‘block of the RF spectrum’ containing all frequencies from DC to 50kHz, from both the ‘wanted’ and ‘unwanted’ mixes with the LO.

This can be referred to as ‘base-band’ frequencies. Some obviously within the normal hearing range, however all signals are present simultaneously, therefore would appear as a garbled mess!

This forms the basis of the simple ***SDR*** RX.

If you were to feed this base-band ‘block’ into a spectrum analyser, this would display all of the signals present within the block as discrete signals, i.e. a simple band scope (ignoring the image problem).

Most people do not have spectrum analysers just sitting about in their shack doing nothing!

However there are a lot of PC’s with a soundcard in shacks that can be used!

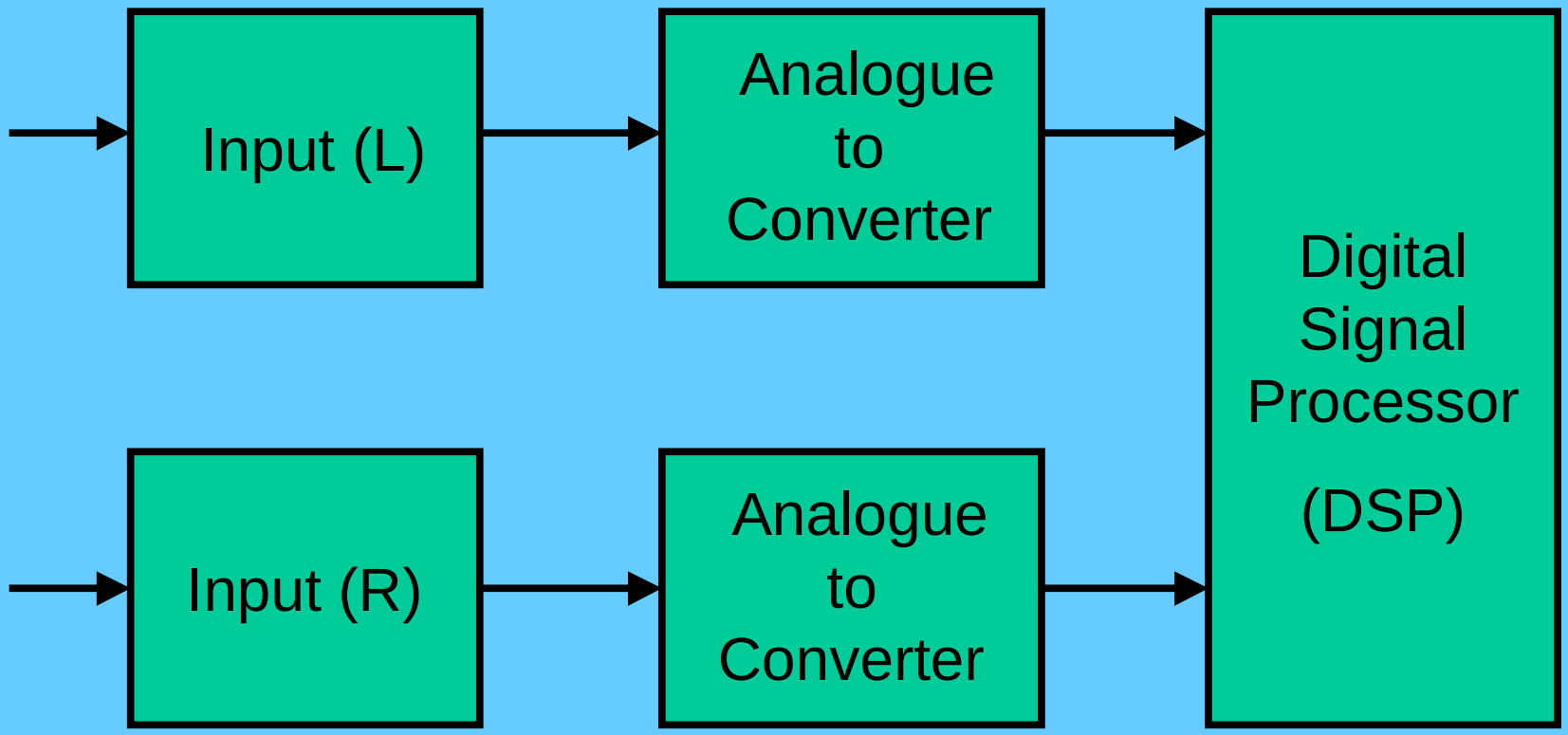
This is the point where we start to leave the simple ***D-C*** receiver behind.

What does a sound card actually do?

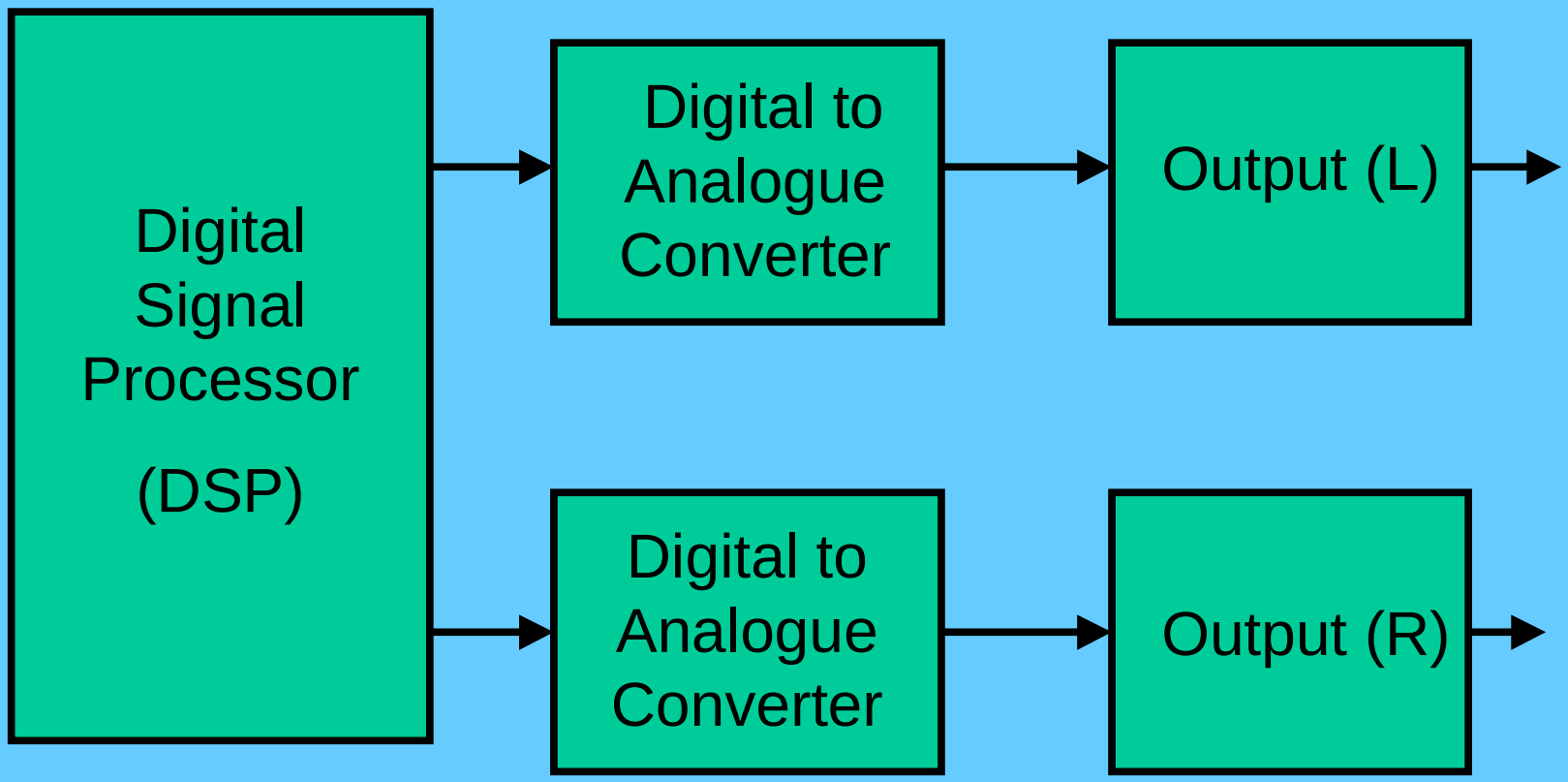
We are feeding in a base-band signal block containing frequencies from DC to 50kHz. Computers only understand (questionable)! digital signals comprising ‘0’s and ‘1’s.

The first stage is to ‘digitise’ this base-band block.

PC Sound Card (Input)



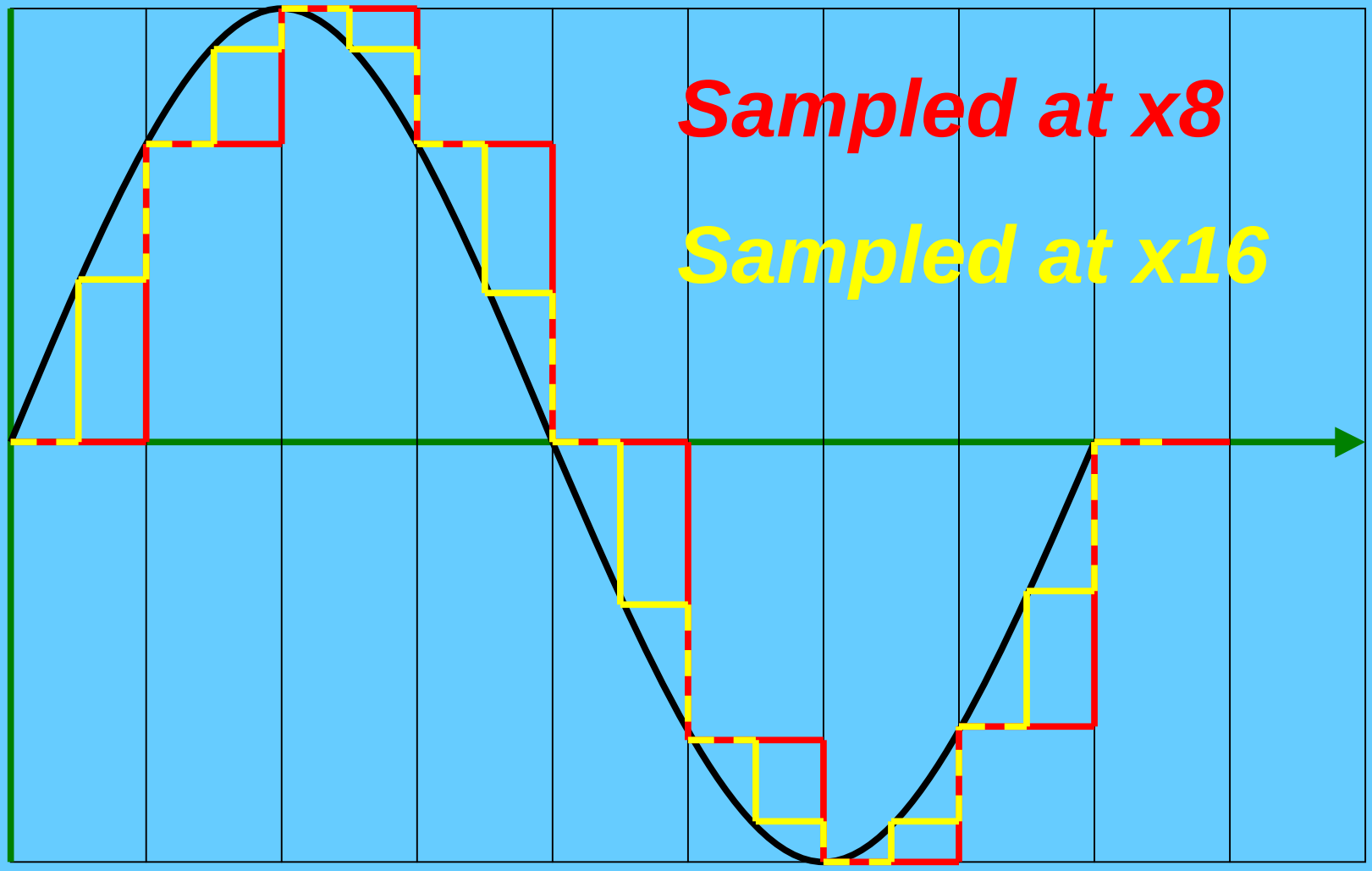
PC Sound Card (Output)



Analogue to Digital Conversion

The basic principal of A/D conversion is to take an instantaneous ‘snap shot’ or ‘sample’ of the amplitude of the incoming analogue signal, and represent that as a numeric value at the output.

In order to accurately represent the incoming signal in numeric form, these samples must be taken at a rate greater than the highest frequency required.


Sampled at x8
Sampled at x16

The previous example showing sampling at two different rates is very simplistic.

The sampling is in-phase with the input waveform, i.e. first sample exactly as the waveform crosses the zero point, and an exact multiple of the frequency.

In reality the waveform being sampled is not a simple sine wave, but a complex waveform containing frequencies from DC to 50kHz.

Most standard PC soundcards allow the incoming signal to be sampled at rates up to 48kHz per channel. High end professional soundcards allow higher sampling rates, typically 96kHz and above.

It can be seen that higher sampling rates will allow a higher input frequency to be digitised.

Typically, frequencies up to approximately half the sampling rate can be reproduced.

This is determined by the ***Nyquist*** sampling theorem which defines the minimum rate needed to reproduce the input waveform.

Sampling a 1kHz input signal at 48kHz would have 48 separate digitised values associated with it, but a 24kHz input signal just 2 samples.

A second parameter associated with A/D conversion, is the number of bits in the output word or resolution.

Most common soundcards operate using 16 bit A/D converters, that is the output can be 1 of 65536 discrete values.

High end soundcards use 24 bit A/D converters allowing a greater range of output values (smaller steps).

From the above it can be seen that for optimum results it is desirable to use the highest sampling rate and maximum bit resolution.

For interest, one limiting factor of using the PC soundcard A/D converters is that they are only intended for audio. A few are available at high cost capable of sampling at 192kHz.

Using external hardware, 16 and 24 bit A/D converters sampling at 100 and 250MHz are easily obtained. These potentially allow the entire HF spectrum up to 30MHz to be sampled at one time. This must be the way ahead.

Having said what is most desirable, good results can still be obtained with more modest equipment.

My demonstration tonight is using an on-board sound chip, usually not the highest quality, 44.1kHz or 48kHz sampling and 16 bits.

All of the demodulation is carried out by the PC, using the DSP functions of the soundcard, and a large amount of processing is necessary.

It is desirable to use the fastest possible processor available, this reduces the processing time (delay).

It is possible to run using slower PC's but the quality suffers, my demo use a 600MHz PIII.

There are a number of excellent audio spectrum analyser programs freely available, such as ***Spectran***.

These can provide spectrum and waterfall type displays of the incoming signals

This is all well and good, but still not possible to demodulate the signals with this type of software, not yet a real receiver.

Mathematics! Not my strong point!!

It is difficult not to make comparisons with the analogue world, but the output of the A/D converter is just a high speed succession of discrete numeric values.

This is where the ***SDR*** really begins.

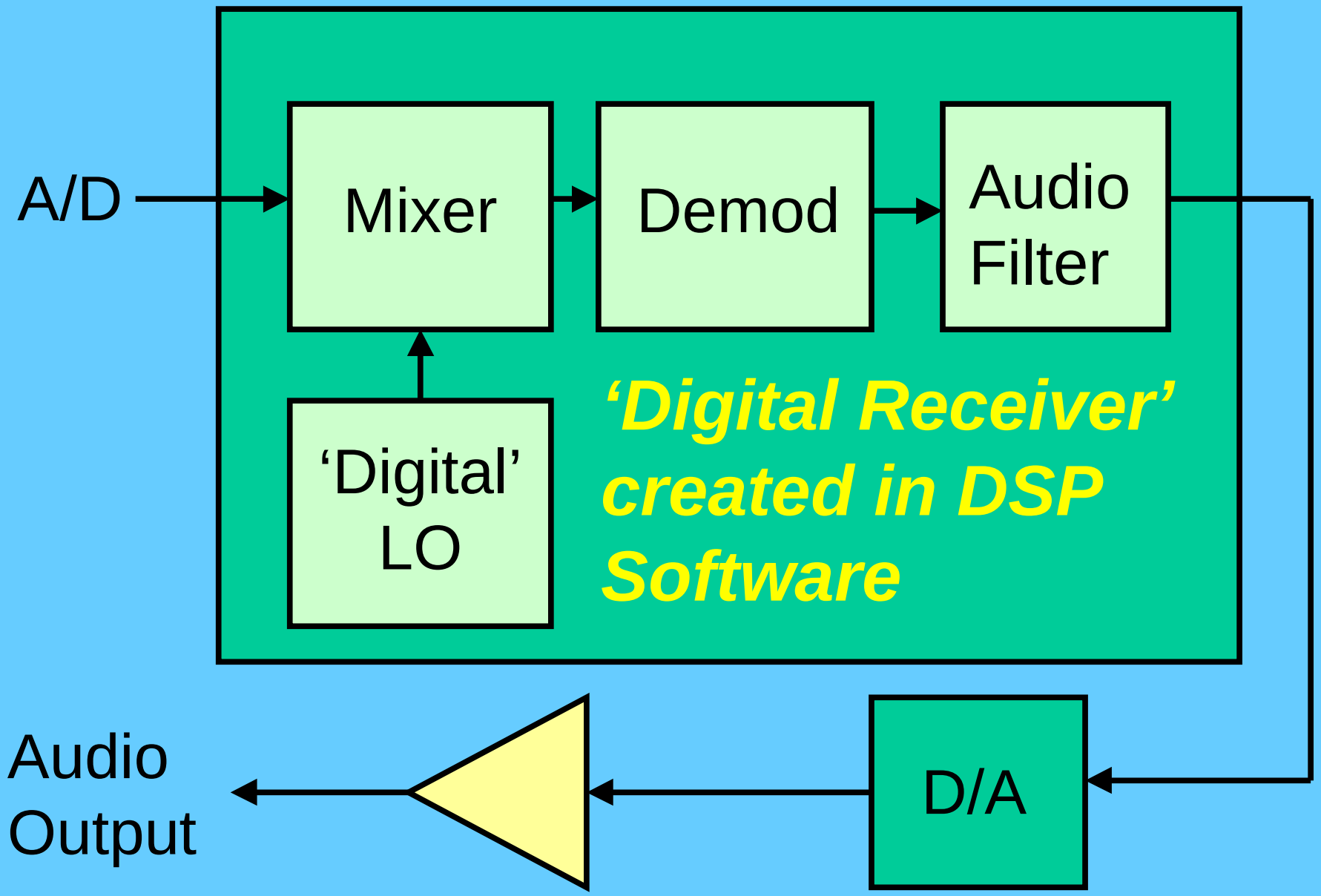
We are all familiar with conventional mixers and their operation. Mixing is really only a mathematical multiplication process. The main products being the ‘sum’ and ‘difference’ of the input signals. This process can be carried out within the DSP programming.

In order to demodulate signals from our base-band block of DC - 50kHz, it is necessary to create a ***‘software direct conversion receiver’*** within the soundcard DSP chip.

If a ‘digital’ local oscillator is produced, covering DC - 50kHz, in steps of 10Hz, this numeric value can be ‘mixed’ with the digitised base-band block to produce the ‘difference’ signal, i.e. the wanted audio output.

This is the simplest form of ***SDR*** receiver, additional audio filtering etc can also be carried out on the PC soundcard.

This ‘receiver’ still has the problem of poor image rejection.


A/D
Mixer
Demod
Audio Filter
'Digital' LO
'Digital Receiver' created in DSP Software
Audio Output
D/A

The next step is to return to the external RF down-converter.

The simple direct conversion receiver is replaced using a ***'quadrature'*** mixing system, or ***Tayloe*** sampling detector.

This process involves splitting the incoming RF into two, each feeding a mixer.

One of these mixers if fed directly from the LO, the other is fed with the LO shifted by 90º.

The two output signals from these mixers are known as:

I = ***In-phase*** (LO direct) - sine of signal

Q = ***Quadrature*** (LO shifted by 90°) - cosine

This type of receiver is commonly known as an ***I/Q*** or phasing receiver.

Designs for analogue ***I/Q*** or phasing receivers have been published in the ***ARRL Handbook*** and ***VHF Communications.***

Subsequent processing of the ***I & Q*** signals in the analogue world is difficult, processing needs one to be shifted by 90º before addition.

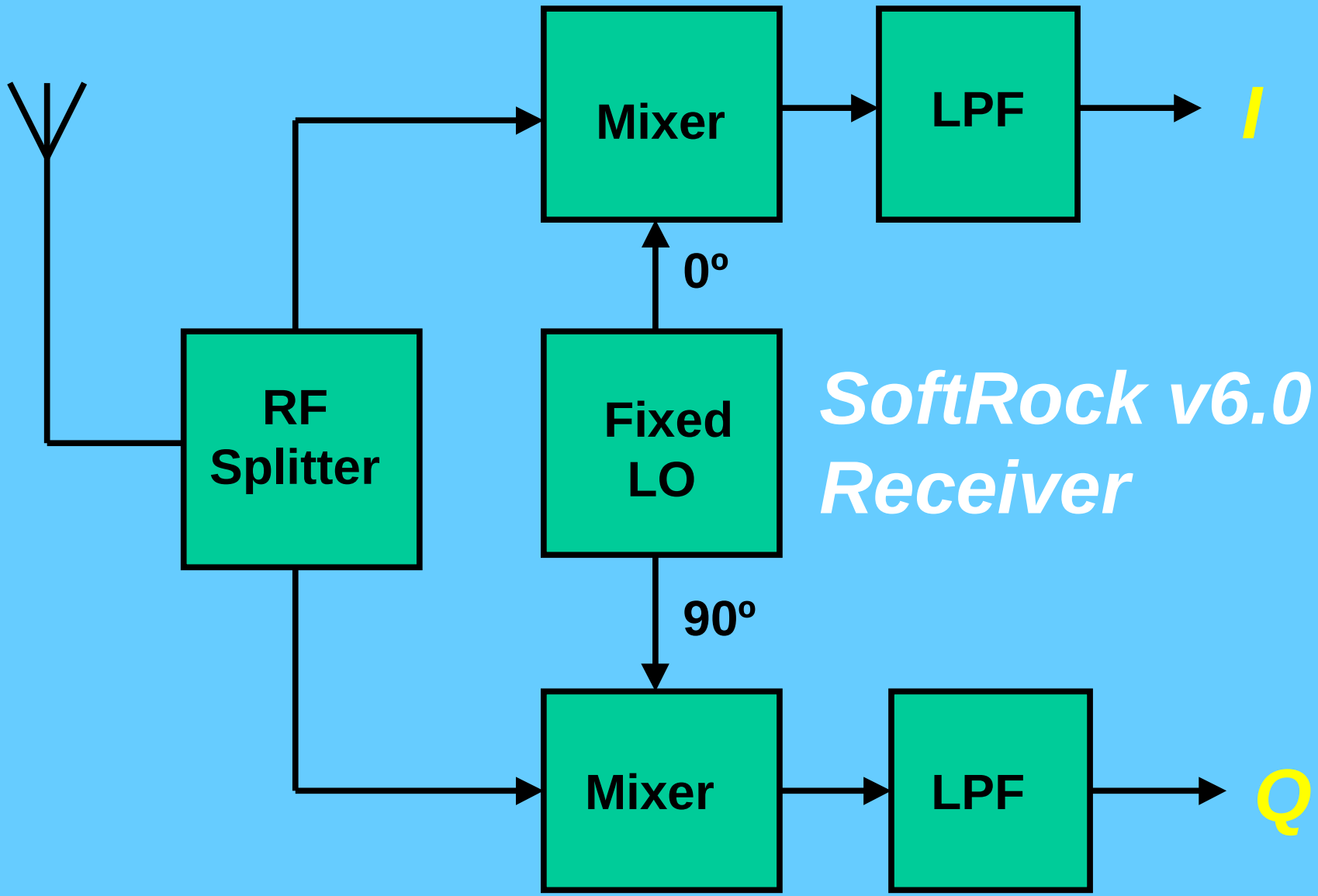
In the analogue world, phase shift is frequency dependant, therefore not possible to keep the phase shift constant across a wide range of frequencies.

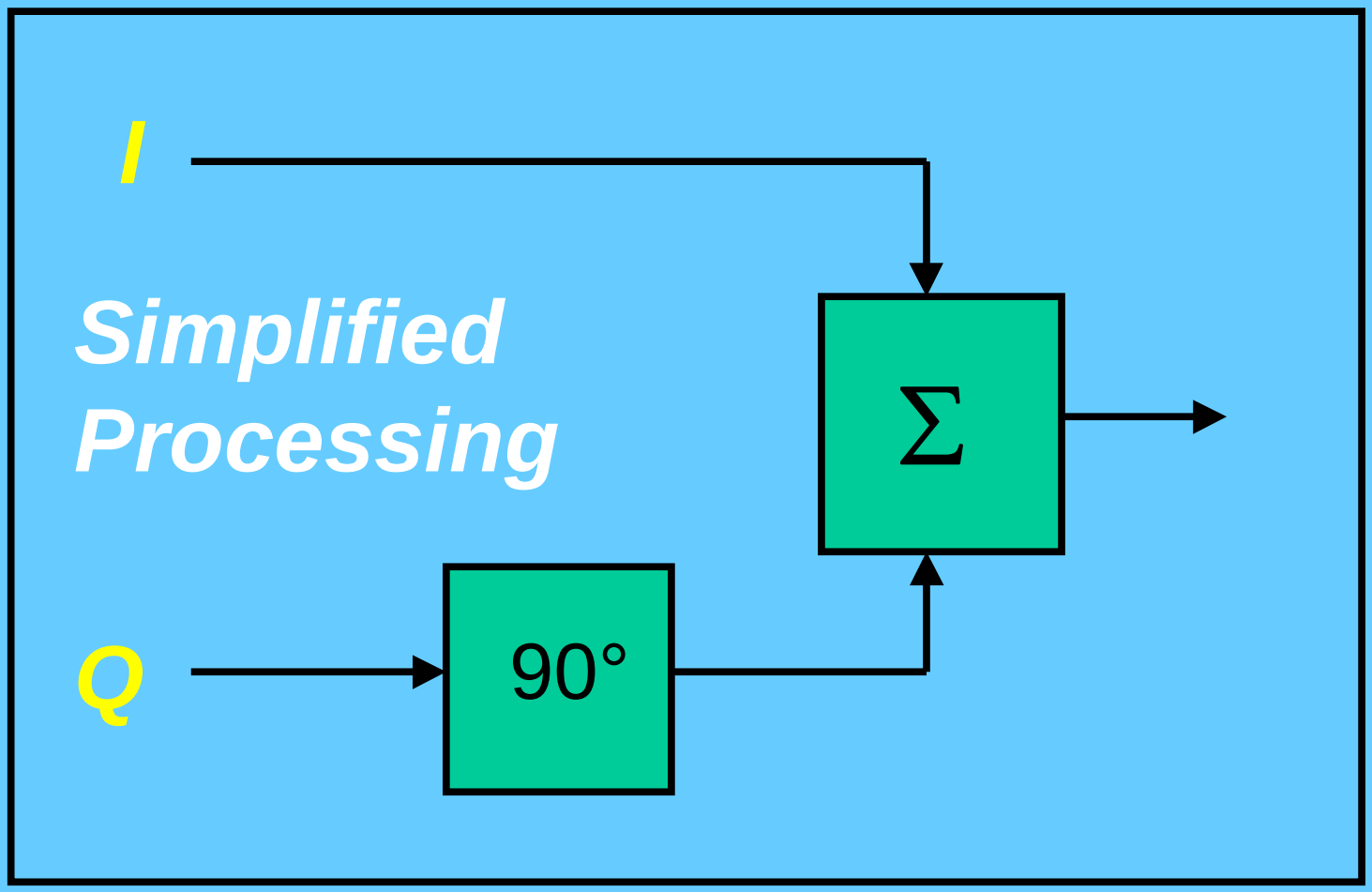
In the digital world, a constant phase shift irrespective of frequency is possible.

This process of phase shifting and cancellation was first used in the generation of SSB signals using the ‘phasing’ method.

This achieved cancellation of the unwanted side-band using the addition of the sine and cosine signals.

This method preceded the now more common way using narrow crystal filters to remove the unwanted side-band.


Mixer
LPF
I
0º
RF
Splitter
Fixed
LO
SoftRock v6.0
Receiver
90º
Mixer
LPF
Q


I
Simplified
Processing
Σ
Q
90°

The two base-band signals ***I & Q*** can be fed to the left and right hand channels of the PC soundcard for subsequent A/D conversion and processing.

By careful processing, and providing the phase and amplitude of these two signals can be closely matched, it is possible to achieve approaching 70dB rejection of the image frequency. This is the first major advantage of the ***I/Q*** receiver over the simple ***D-C*** one.

A number of factors have to be taken into account associated with the soundcard, noise levels, phase, amplitude etc, therefore performance can vary between different PC's and different programs.

I am typically only achieving 40-50dB rejection using one program, and around 30dB with another, using the same hardware.

To obtain 40dB of image rejection, the ***I & Q*** signals arriving at the demodulation stage need to be within 1º phase and 0.1dB relative amplitude.

To achieve 60dB image rejection these need to be within 0.1º phase and 0.01dB of each other.

A traditional superhetrodyne receiver using a crystal filter in the I.F. stages is considerably better than this.

Another concept that is difficult to grasp is that of ***'negative frequency'***. In the 'real' world positive and negative frequencies merge into one, i.e. the 'wanted' signal and image response both appear as the same frequency.

In the 'digital' DSP world, negative frequencies ***can*** exist, therefore it is possible to mathematically process the ***I & Q*** inputs to separate the two signals.

It is therefore possible to ‘unfold’ the image and display the spectrum each side of the LO (zero frequency).

It is by this process that it is possible to obtain a block of frequencies virtually equivalent to the soundcard sampling frequency, yet still satisfying the minimum ***Nyquist*** sampling requirements.

With a soundcard sampling at 48kHz, the spectrum +/- approx. 24kHz can be seen.


Output
Image
Wanted
f
7.000
7.050
LO
7.100
MHz
Wanted & Image
Input
f
DC
Zero
50
kHz

In addition to using the ***SDR*** software for listening to ‘live’ signals, it is possible to record the incoming ***I & Q*** waveforms using any suitable PC sound recording software and store it as a ***.WAV*** file.

The entire spectrum is recorded, therefore when played back, you can tune around the band and operate all features of the ***SDR*** software.

Sampling the stereo input at 44.1kHz / 16 bits requires around 10Mb storage per minute.

The SoftRock v6.0 SDR Receiver

Well that's a look at the concept behind the ***SDR***

The ***SoftRock*** series of simple ***SDR*** receivers have been developed by ***Tony Parks, KB9YIG and Bill Tracey KD5TFD.***

The current range of ***v6.0*** kits are based around a 40/80m receiver, variants are available for 160m and 30m bands. There is also a ***v7.0*** kit for 10m.

You will have probably seen adverts by one UK supplier selling the kits for £33 including p&p (they do include documentation and software on a CD).

On enquiring, they had a waiting list of 20+ people, and thought it would take approx. 3 weeks to supply.

There is a ***much*** easier and ***cheaper*** way!!

Tony Parks, one of the designers, supplies the kits direct. He is very fast to respond to emails, and accepts payment directly to his ***PayPal*** account ***raparks@ctcisp.com***

The ***v6.0*** kits are $15 and the v7.0 kit $19 including postage outside the USA.

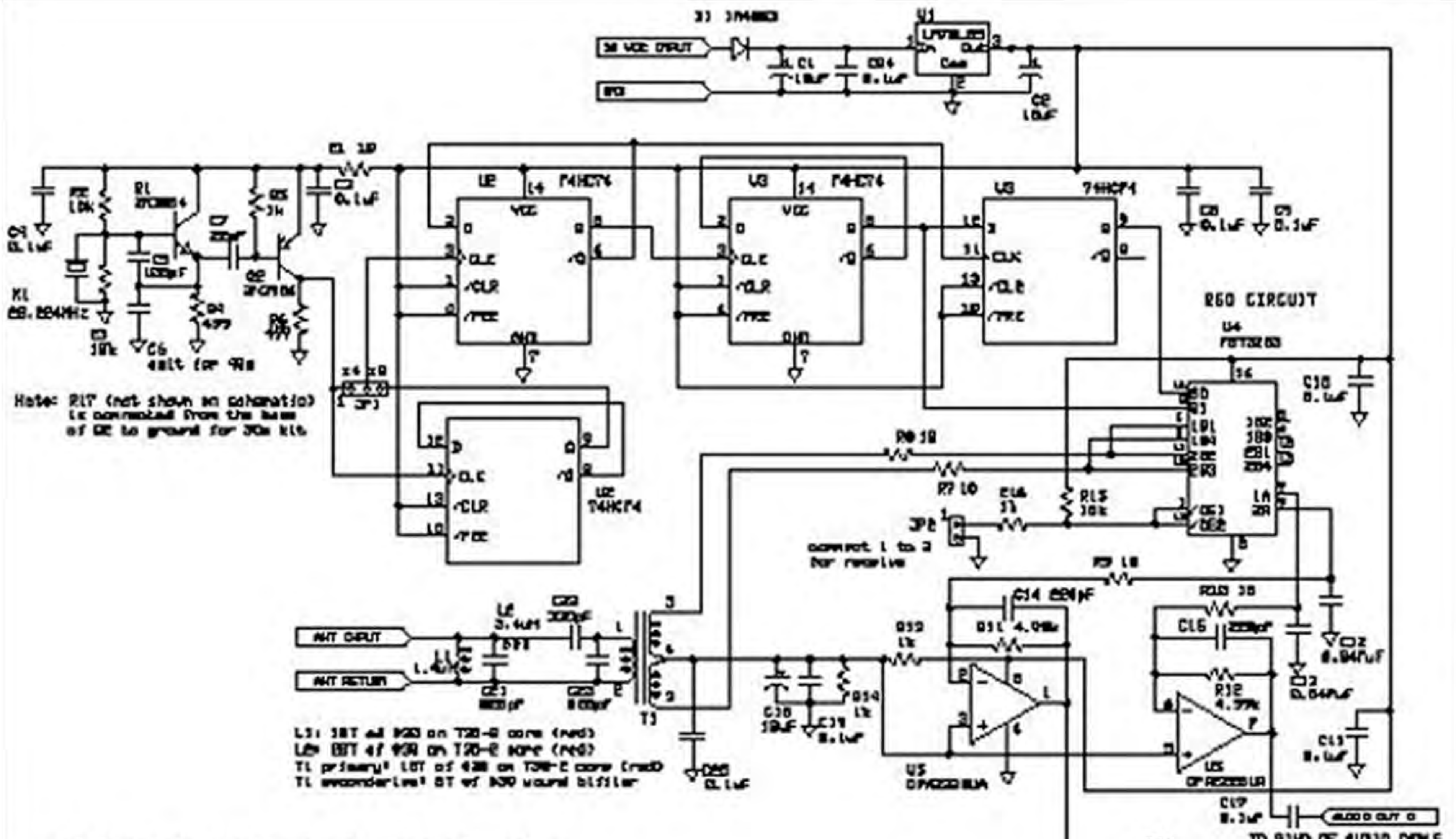
My first purchase took 8 days from initial email enquiry to arriving in the post, cost £19.63 for a ***v6.0*** and ***v7.0*** kit.

My second order took 7 days, 2 x ***v6.0*** kits £16.62.

So for about £3 more than the UK source, I have 4 receivers, and in less time!

All of the documentation, circuit diagrams, board layouts, construction details and bill of materials can be downloaded from the internet for free.

All of the available software packages are also available for free.



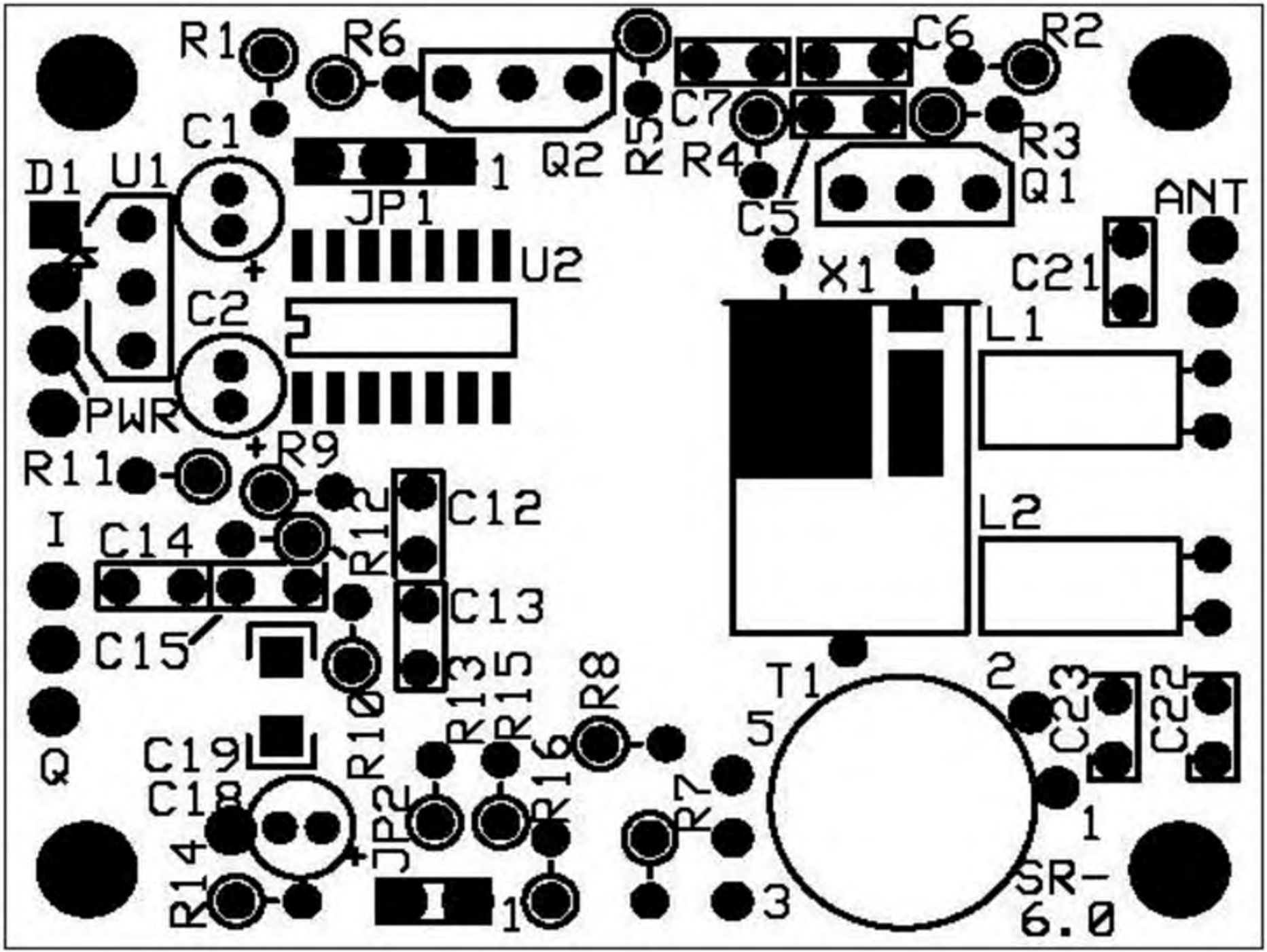
The schematic shows 40m/80m component values.

The 160m component values	The 30m component values
X1 14.784 MHz crystal	X1 40.0 MHz crystal
C5 220 pF	C5 100 pF
C6 150 pF	C6 not used, R17 22.1k added in C6-C7 location
C21, C23 6800 pF	C21, C23 470 pF
C22 470 pF	C22 100 pF
L1: 16T of #30 on T30-2 core (red)	L1: 14T of #30 on T30-2 core (red)
L2: 8TT of #30 on T30-2 core (red)	L2: 22T of #30 on T30-2 core (red)
T1 primary: 16T of #30 on T30-2 core (red)	T1 primary: 14T of #30 on T30-2 core (red)
T1 secondaries: 7T of #30 wound bifilar	T1 secondaries: 7T of #30 wound bifilar

FLEX RADIO FRIENDS

SoftRock_v6.0 RX

Tony KB9YIG — Rev 0.0 — 03/29/2006 — Page 1 of 1


R1
R6
Q2
R5
C6
R2
C7
R3
R4
Q1
C1
D1
U1
JP1
1
C5
X1
ANT
C21
U2
C2
L1
PWR
R11
R9
C12
I
C14
R12
L2
C13
C15
R13
R15
R8
T1
2
C23
C22
C19
R10
R16
5
Q
C18
JP2
R7
1
R14
3
SR-
6.0

Band Coverage

SoftRock v6.0		Sampling
40m	7.032 - 7.080 MHz	48 kHz
	7.008 - 7.106 MHz	96 kHz
80m	3.504 - 3.552 MHz	48 kHz
	3.480 - 3.576 MHz	96 kHz
160m	1.819 - 1.867 MHz	48 kHz
	1.795 - 1.891 MHz	96 kHz
30m	10.100 - 10.147 MHz	48 kHz

Band Coverage

SoftRock v7.0		*Sampling*
10m	28.036 - 28.084 MHz	48 kHz
	28.012 - 28.108 MHz	96 kHz
10m	28.200 - 28.248 MHz	48 kHz
	28.176 - 28.272 MHz	96 kHz

Kit is supplied with two crystals giving centre frequency (LO) either:

28.060 or 28.224 MHz

Uses?

A cheap, but powerful, simple single or dual band HF receiver when used in conjunction with a PC.

A companion RX to a simple QRP transmitter.

As a band-scope, or convert a 30m ***v6.0*** to 10.7MHz to monitor the I.F. from a RX, i.e. IC-R7000, would give you ***SDR*** facilities covering 25MHz - 2GHz.

Use a ***v7.0*** with VHF to 28MHz converters.

For my microwave work, use a 144MHz to 28MHz receive converter on the output of the transverters for any band (1.2, 2.3, 3.4, 5.7, 10 & 24GHz).

At present need to tune to find signal and align the dish at the same time. Should be able to ‘see’ a carrier / weak morse signal on the waterfall regardless of frequency, therefore only need to peak the dish.

What do you get for your money?

A good quality double sided, silk screen printed, solder resist coated PCB.

All board mounted components.

Note: there are 11 surface mounted capacitors and 4 SMD ic’s, the rest are conventional leaded components.

Band specific parts, crystal, toroidal cores and enamel wire is also supplied.

The only additional items the builder needs to supply are a suitable box, aerial input connector and a soundcard lead.

The unit requires a 9-12v DC supply, and draws about 35mA.

Building the receiver is straight forward, and the builder's notes are thorough and easy to follow, and should take a couple of hours.

The only potential difficulties for constructors are:

1) ***Surface mounted devices***

2) ***Winding the coils***

3) ***Identifying resistors***

Surface Mounted Components

There are 11 capacitors and 4 ic's in the kit.

All capacitors are the same value, 1206 size.

All ic's have unique positions. 3 types have different number of pins. Pin 1 marked on pcb.

Need a fine tipped soldering iron, thin solder (<0.5mm), and possibly optical assistance!

Note: a larger iron might be needed to solder leads to the pcb ground plane.

Winding Coils

There are 2 inductors and a transformer to wind.

The inductors are wound on a ferrite toroidal core, remember each time the wire passes through the core is 1 turn. Try to space the turns evenly around the core.

The enamel wire supplied is not ‘self fluxing’ when heated. It is necessary to remove it either by scraping or using emery paper.

The primary of the transformer is wound in a similar fashion to the inductors. The start and finish of the winding should be in a similar position.

The secondary is ***'bifilar'*** wound. A length of wire is folded in half, then twisted using a small pin vice or drill. Aim for about 2 - 3 turns per cm.

The secondary is wound with this twisted wire starting 180º from the primary.

The documentation is not clear, but the winding should be in the same direction as the primary and start and finish 180º from the start and finish of the primary.

Unwind the ends of the secondary and remove the enamel, best to tin them as well.

Identify 1 winding using a ohm meter, identify the start, then connect the end of the first winding to start of the second winding. This produces a centre tapped transformer.

The transformer was the only problem I had with the first receiver built.

I did not fully clear off the enamel on one of the secondary leads, although it appeared to be tinned, in fact solder stuck to enamel.

On running the receiver performance was poor, little or no image rejection, but fault easily identified doing resistance checks on the transformer in-circuit.

In the ***SoftRock v7.0*** receiver, the transformer is a little more complicated.

There are 2 centre tapped secondary windings. Four pieces of wire are twisted together to form the bifilar winding.

Each winding is separated, and the process of making the centre tapped winding is carried out twice.

Resistors

The kit uses 4 band 1% tolerance resistors. Colour band markings (brown / red) difficult to identify.

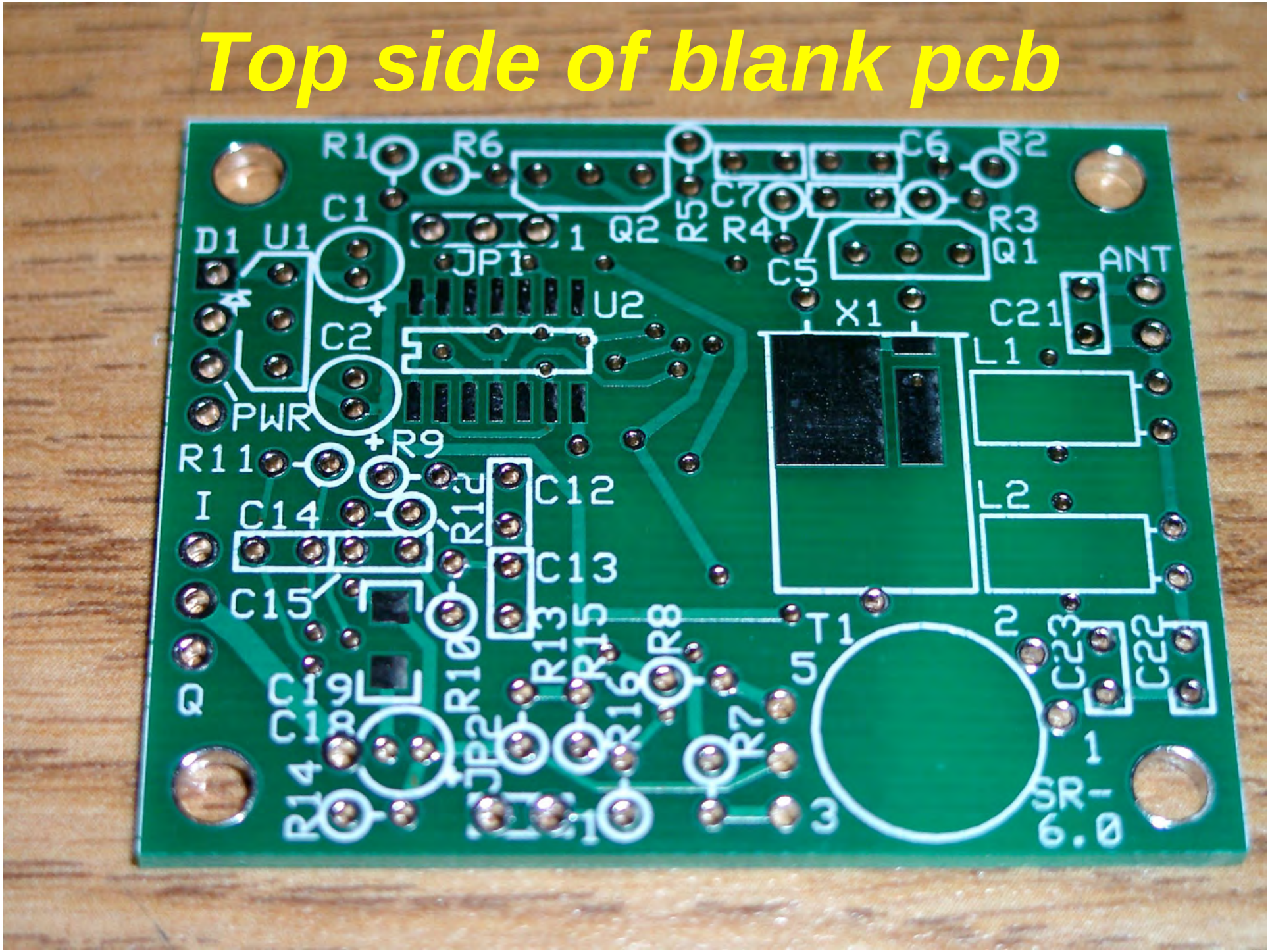
Use an ohm meter to confirm values.

The following slides show one of the ***SoftRock v6.0*** kits in various stages of construction from the blank pcb to completion.

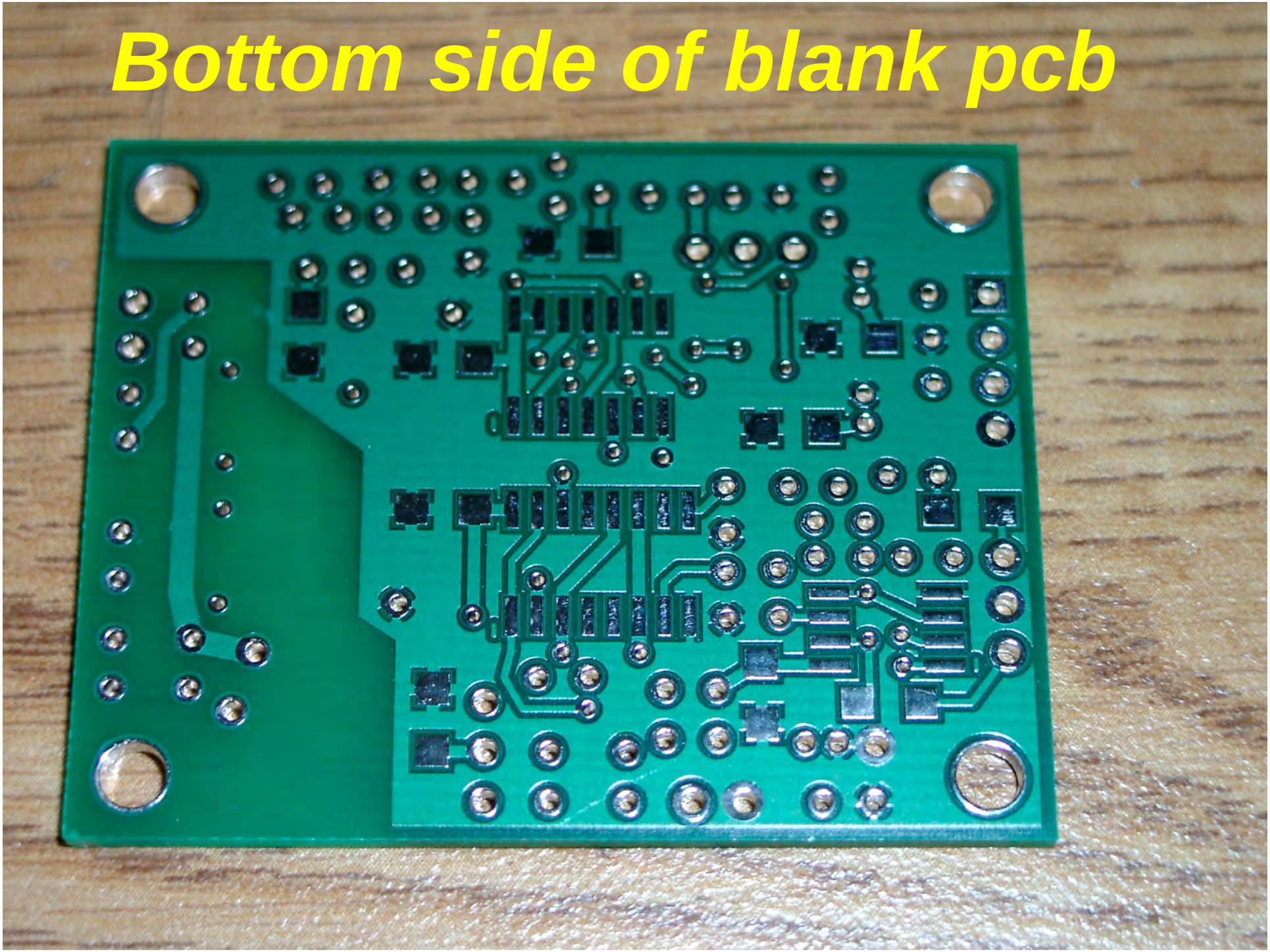

JP1 1-2 7.056 MHz
JP1 2-3 3.528 MHz
The Original Celebrated CURIOUSLY STRONG PEPPERMINTS
ALTOIDS
MADE IN GREAT BRITAIN
SoftRock v7.0 10m
SoftRock v6.0 40/80m
X1 40.5 crystal
C5 100 pF
C6 not used, R17 22.1k added in C6-C7 location
C21, C23 470 pF
C22 180 pF
L1: 14T of #30 on T30-2 core (red)
U5
OPA2228UA
TO RING OF AUDIO CABLE
TO TIP OF AUDIO CABLE
RADIO FRIENDS
SoftRock_v6.0 RX
Tony KB9YIG
Rev 0.0
03/29/2006
Page 1 of

Kit of Parts

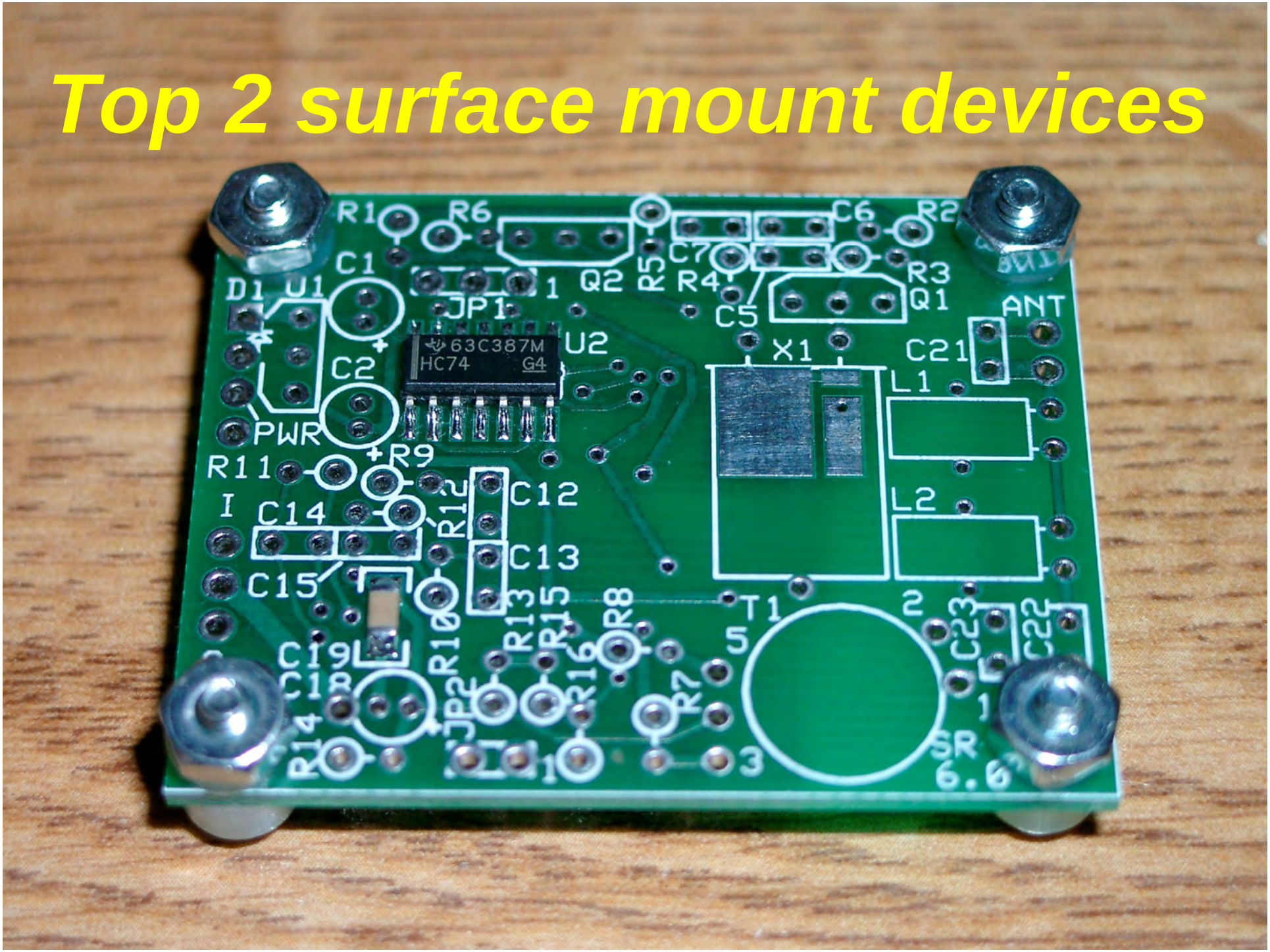
Top side of blank pcb

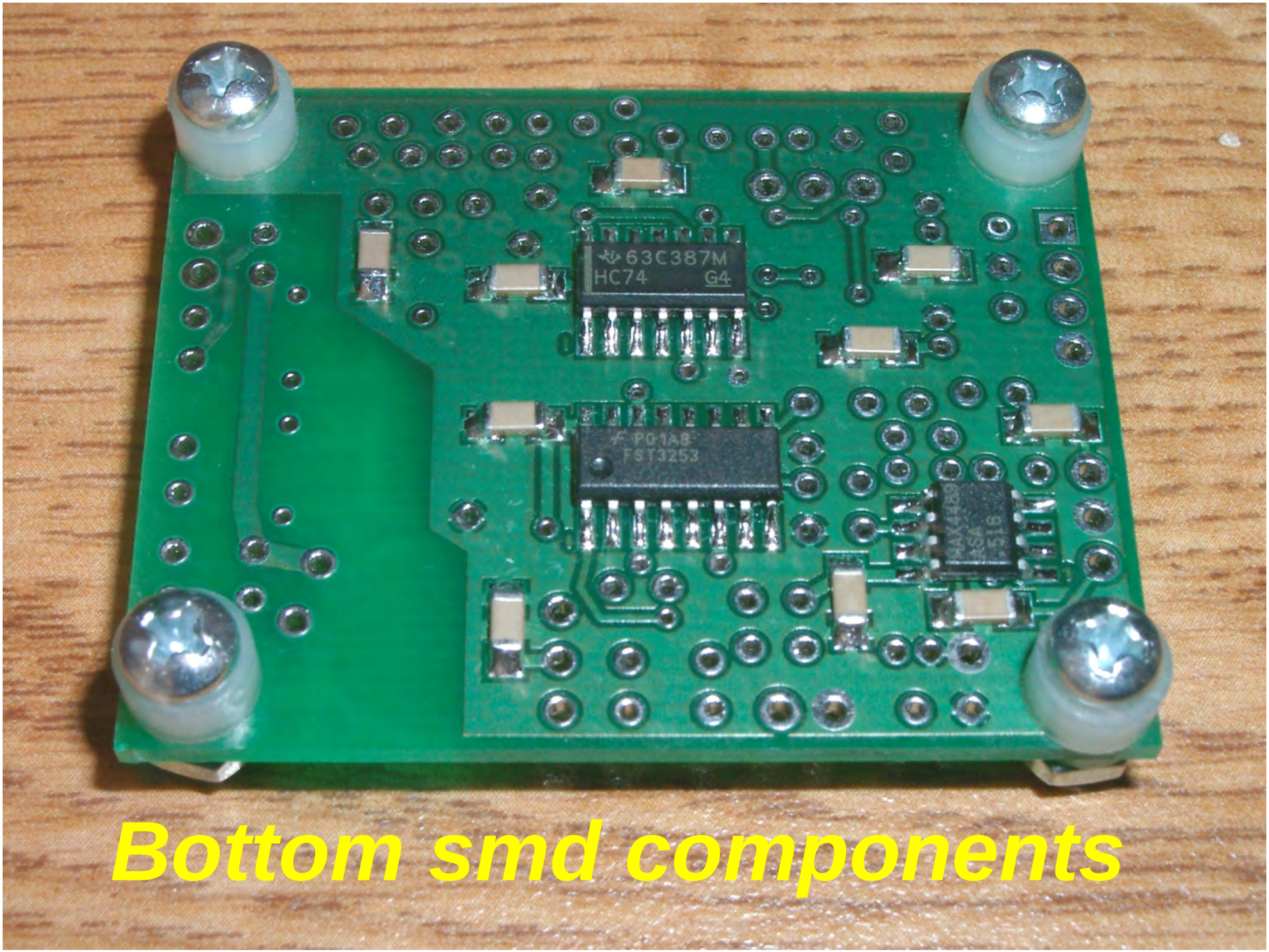


Bottom side of blank pcb



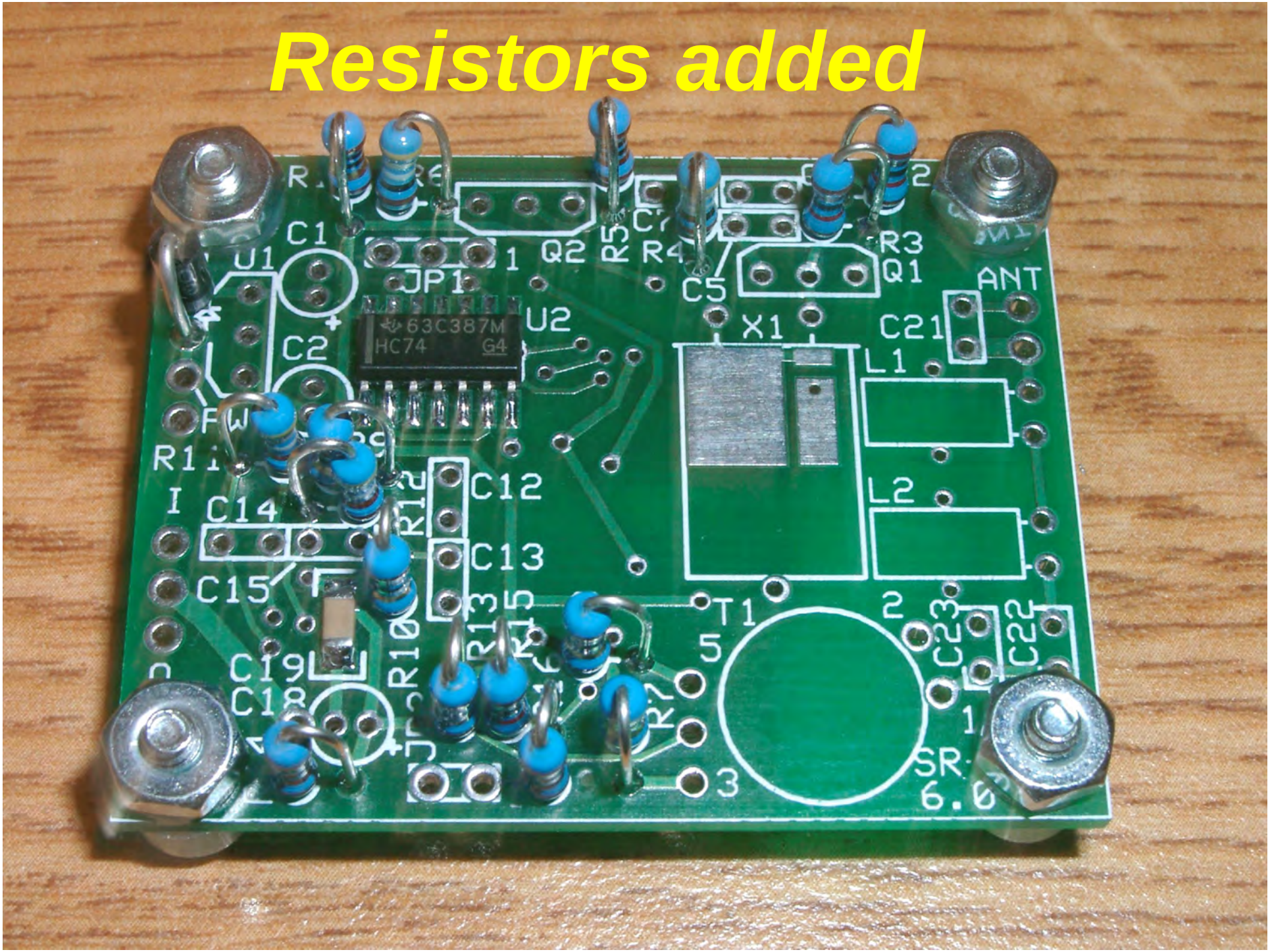
Top 2 surface mount devices

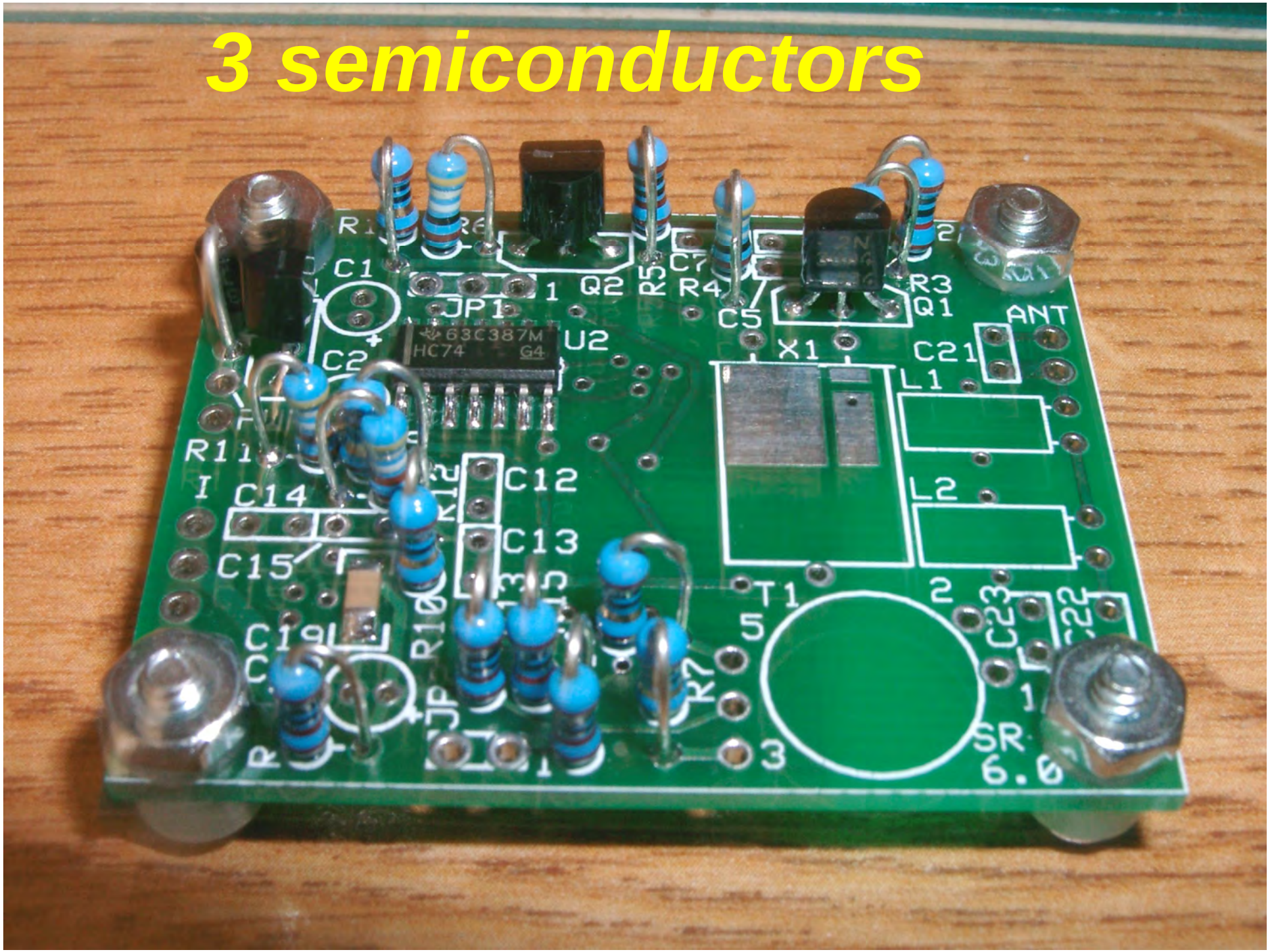




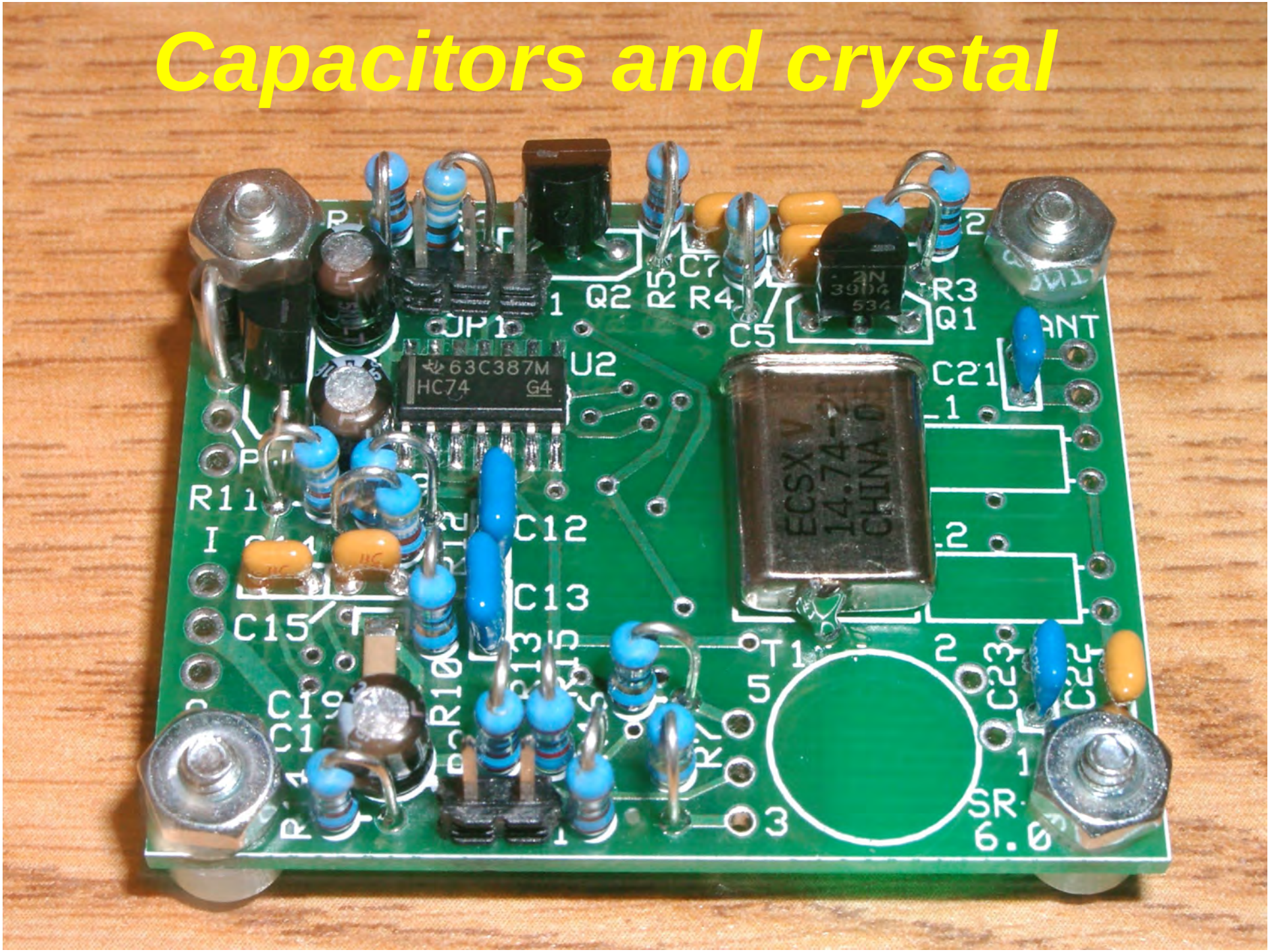
Bottom smd components

Resistors added

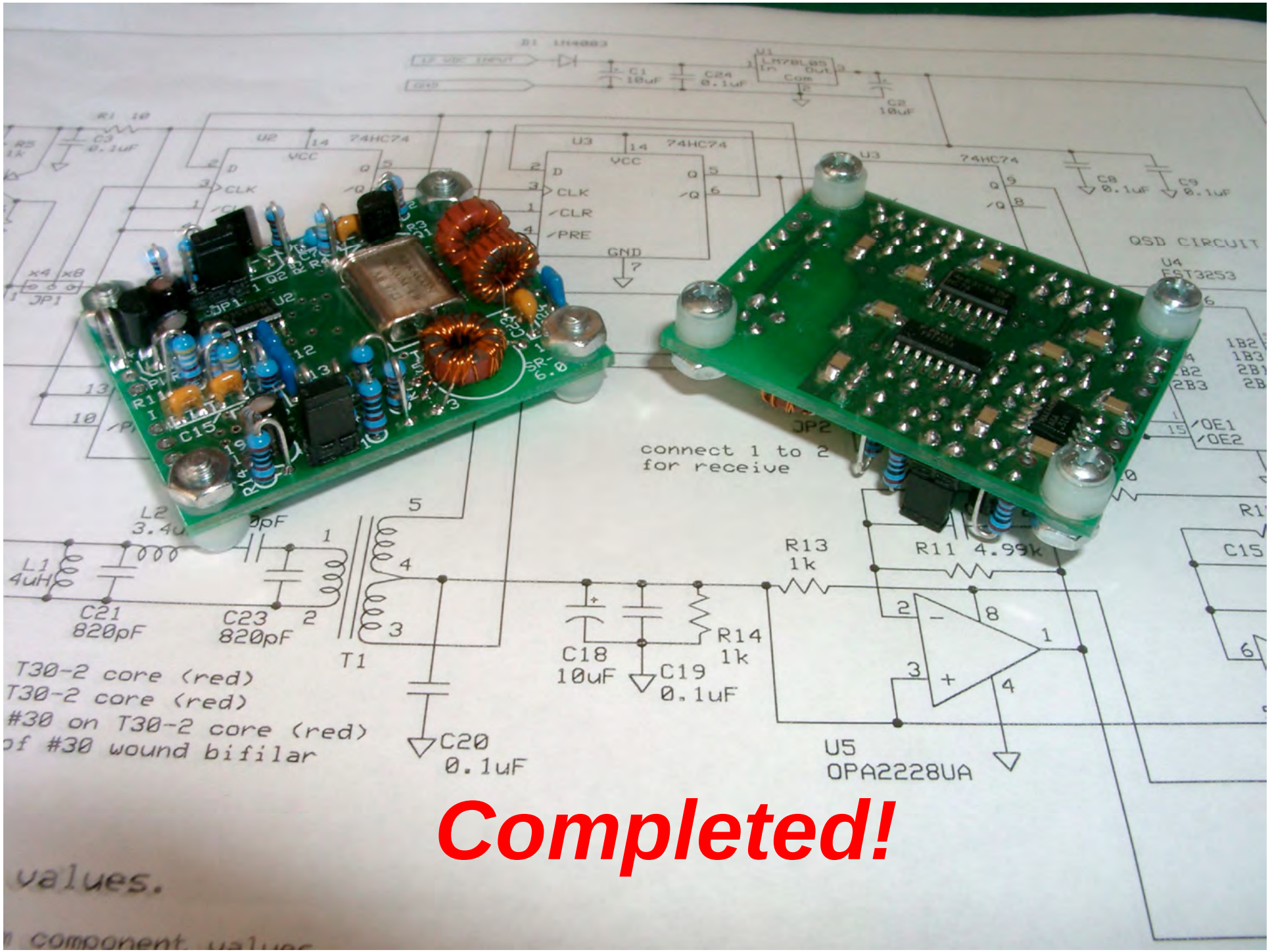


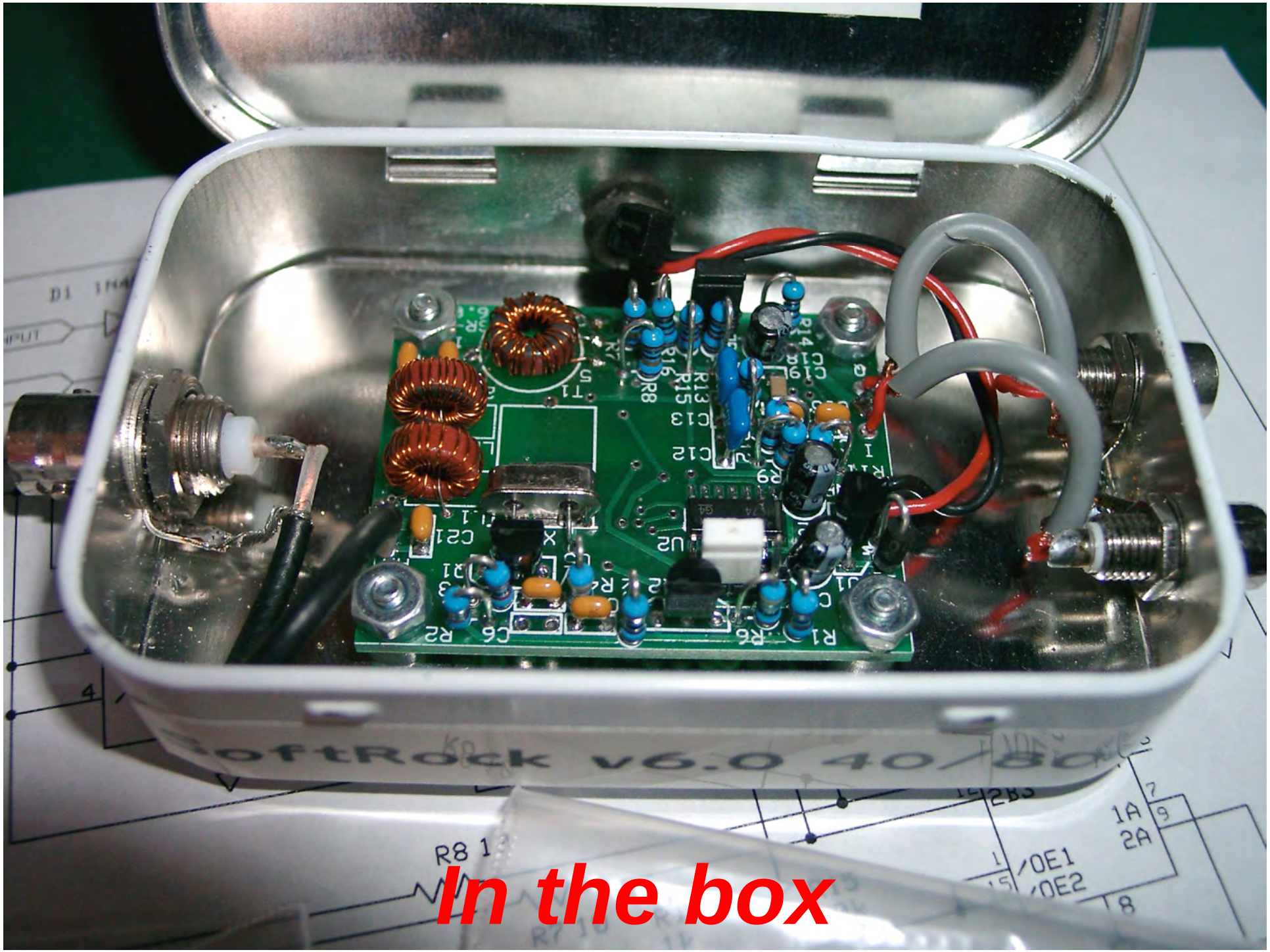

3 semiconductors
C1
C2
JP1
Q2
Q1
R3
R4
R5
C5
U2
X1
C21
ANT
L1
L2
R11
C14
C12
C13
C15
T1
C23
SR
6.0

Capacitors and crystal



Finally the coils


connect 1 to 2
for receive
QSD CIRCUIT
T30-2 core (red)
T30-2 core (red)
#30 on T30-2 core (red)
of #30 wound bifilar
values.
component values
C21
820pF
C23
820pF
T1
C18
10uF
C19
0.1uF
C20
0.1uF
R13
1k
R14
1k
R11 4.99k
U5
OPA2228UA
L1
4uH
L2
3.4u
74HC74
JP1
JP2
Completed!



In the box


SoftRock v7.0 10m
CALLARD & BOWSER
The Original Celebrated
CURIOUSLY STRONG
PEPPERMINTS™
ALTOIDS®
MENTHA PIPERITA
MADE IN GREAT BRITAIN
Ent. Stationer's Hall
SoftRock v6.0 40/80m
ANT INPUT
ANT RETURN
L1
1.4uH
C21
820pF
C23
820pF
C18
C20
0.1uF
R14
R15
10k
R9 10
C14 220pF
R11 4.99k
1k
U5
OPA2228UA
L1: 18T of #30 on T30-2 co
L2: 28T of #30 on T30-2 co
T1 primary: 18T of #30 on T30-2 core (red)
T1 secondaries: 8T of #30 wound bifilar
Ready to go!

SDR Software

There are a number of different programs available that can be used with the ***SoftRock*** receivers.

As previously mentioned, performance is determined by the quality of the soundcard, and the processor speed.

I have found a number of problems using my 600MHz PIII and ***Win98se***.

All of the software currently available is ***'open source'***.

The source code is made freely available to allow anyone with the required skills and tools to modify, adapt or add to the software. This means the programs continue to evolve.

There are very active internet discussion groups for both the software and different hardware.

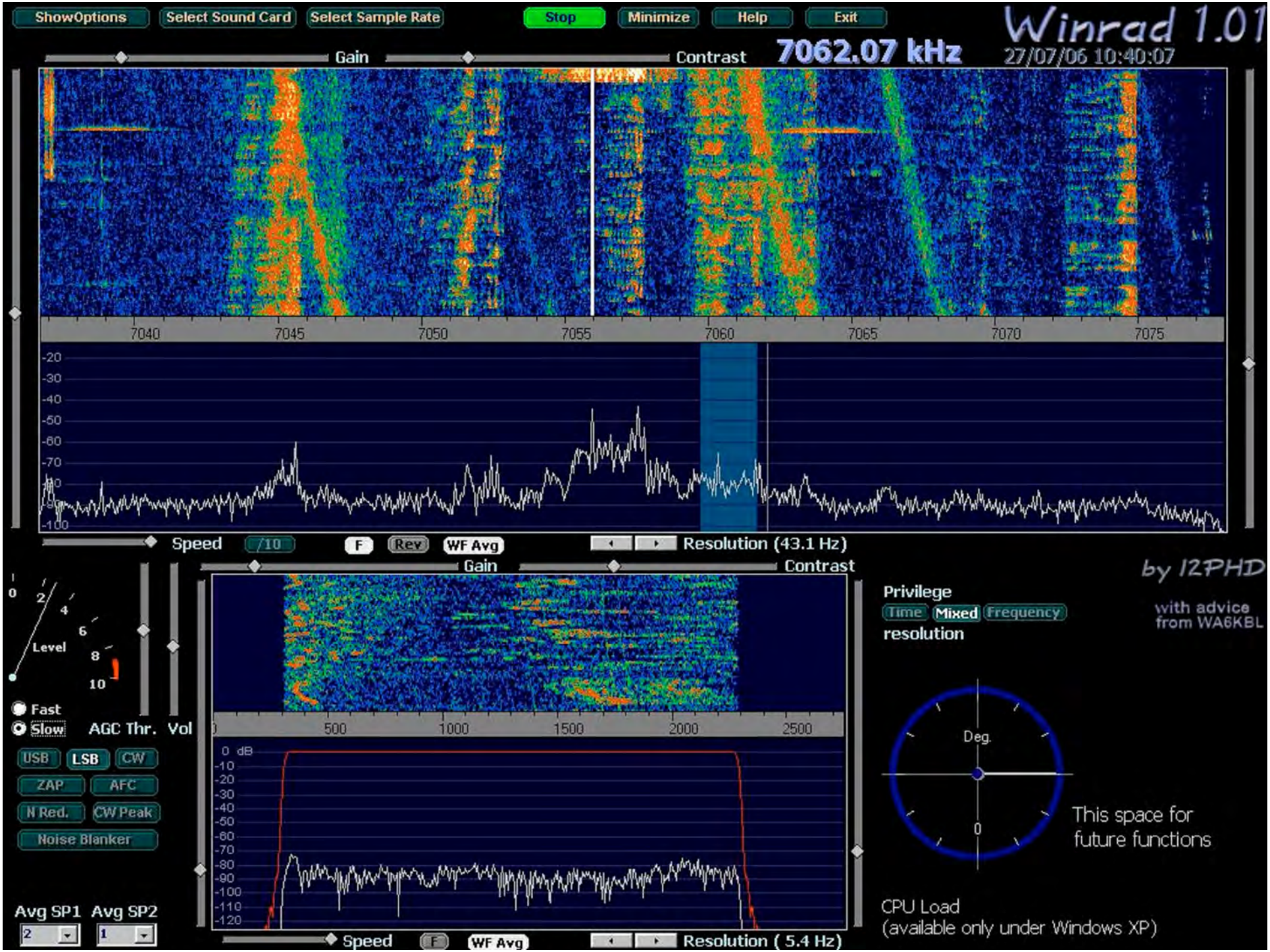
Several of the programs comment that they are unlikely to run using ***Win98***, and appear to be geared more towards ***Win 2000*** or ***XP***.

Initially I could only get 2 programs written by ***Alberto, I2PHD*** to run:

SDRadio Basic ***SDR*** but supports AM / FM

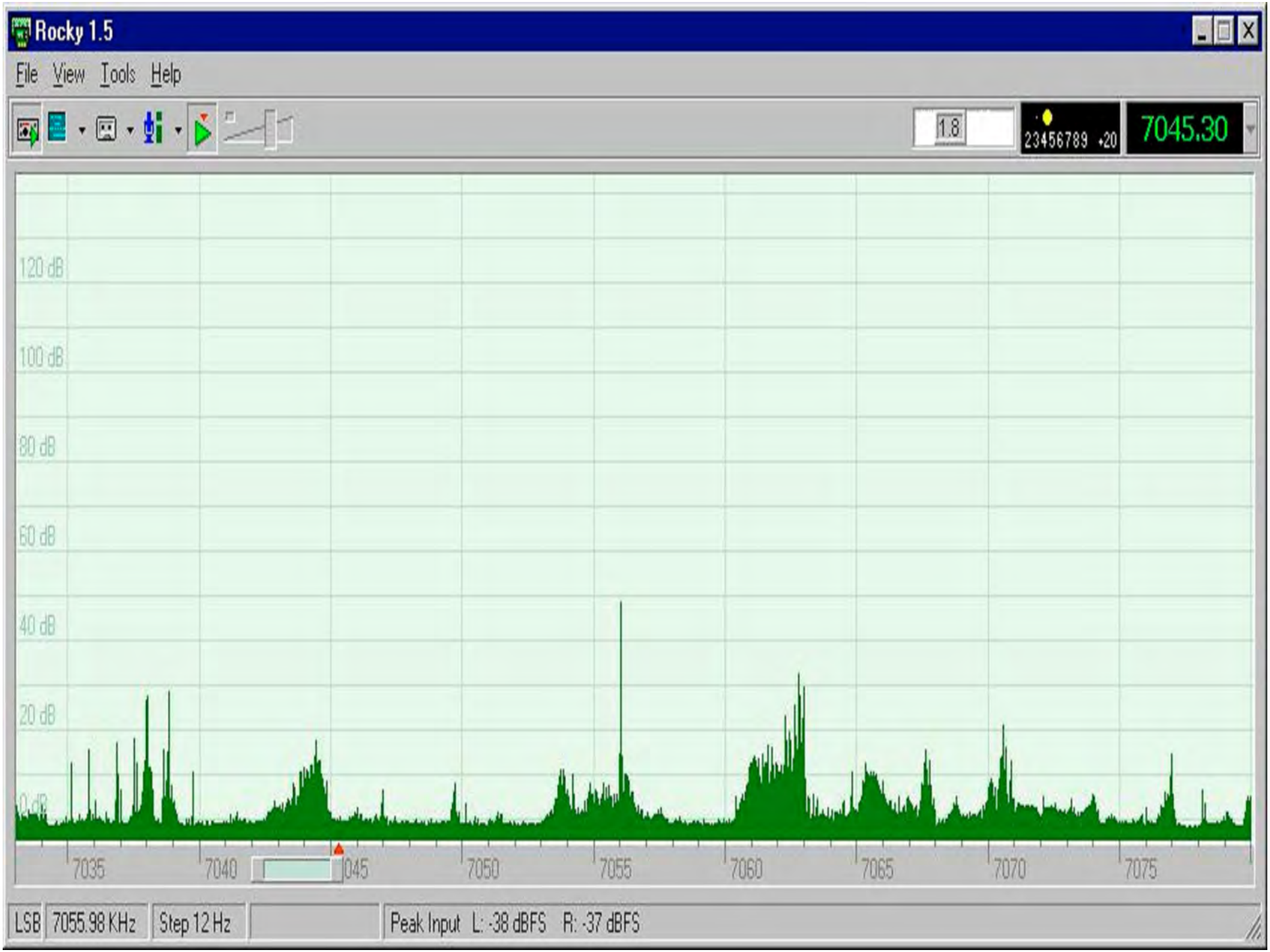
WinRad Designed to look like the ***Linux Linrad*** program.

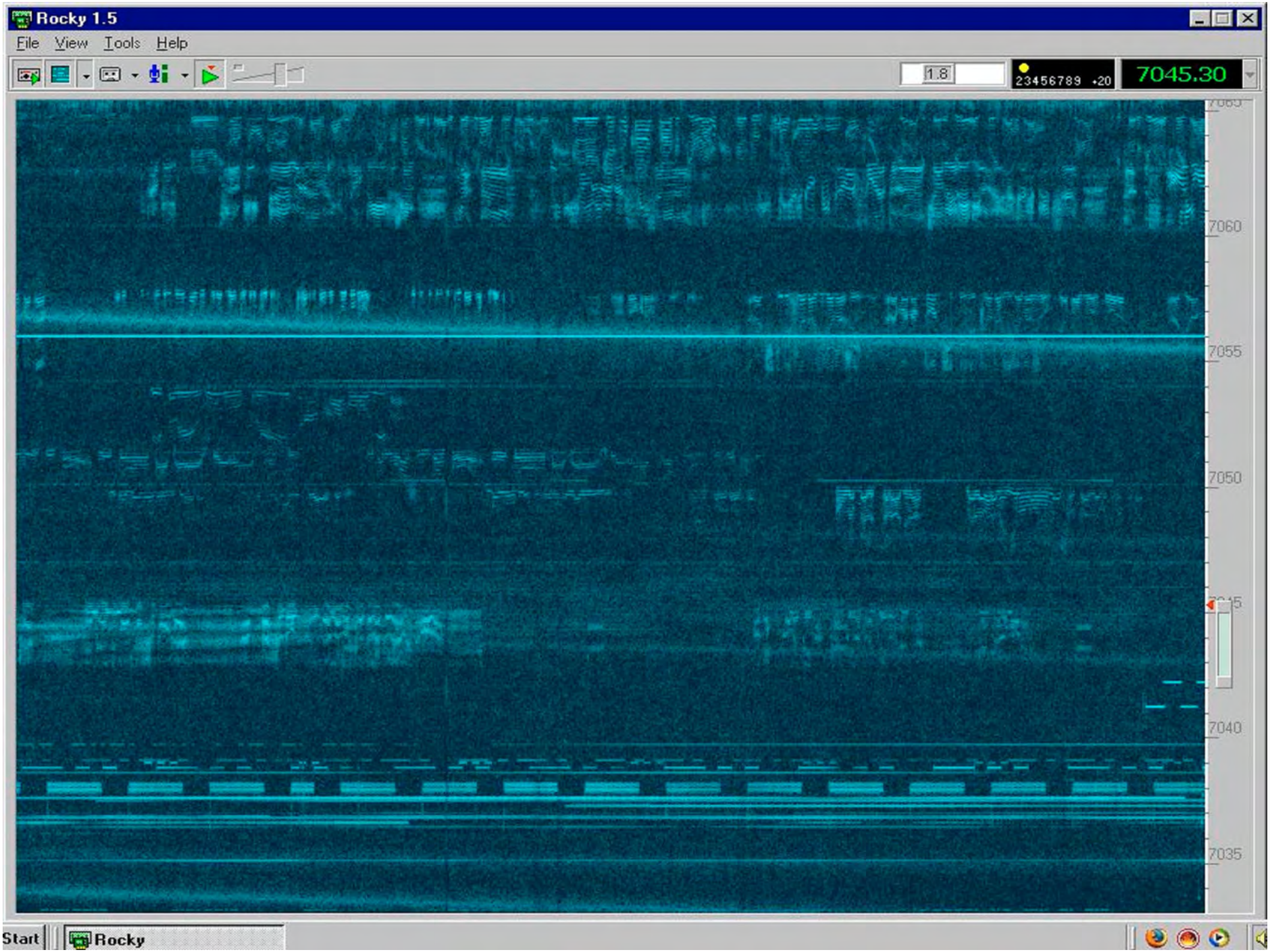

SDRadio
Version 0.99
21:22:07
? Help
_ Minimize
X Exit
Options
7072.38 kHz
by Alberto I2PHD
0 dB
-10
-20
-30
-40
-50
-60
-80
7050
7055
7060
7065
7070
AGC Gain
Lo
Mid
Hi
Denoiser
Rx
St.By
48 kHz
24 kHz
12 kHz
6 kHz
3 kHz
Freq. span
Fine Tune
+0 Hz
Phase delta
0
Level
AM
ECSS
USB
LSB
FM


ShowOptions
Select Sound Card
Select Sample Rate
Stop
Minimize
Help
Exit
Winrad 1.01
7062.07 kHz
27/07/06 10:40:07
Gain
Contrast
7040
7045
7050
7055
7060
7065
7070
7075
-20
-30
-40
-50
-60
-70
-100
Speed
/10
F
Rev
WF Avg
Resolution (43.1 Hz)
Gain
Contrast
by I2PHD
with advice from WA6KBL
Privilege
Time
Mixed
Frequency
resolution
0
2
4
6
8
10
Level
Fast
Slow
AGC Thr.
Vol
USB
LSB
CW
ZAP
AFC
N Red.
CW Peak
Noise Blanker
0
500
1000
1500
2000
2500
0 dB
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
-110
-120
Deg.
0
This space for future functions
Avg SP1
Avg SP2
2
1
Speed
F
WF Avg
Resolution (5.4 Hz)
CPU Load
(available only under Windows XP)

After a little more reading, I managed to get the ***Rocky 1.5*** software by ***Alex VE3NEA*** running. This is the program recommended for initially testing the ***SoftRock*** receivers.

The program did not recognise or allow the soundcard to be selected. The software interfaces with the soundcard using the ***'WMA'*** codec, a quick search on the internet found an updated driver for the on board sound chip.


Rocky 1.5
File View Tools Help
1.8
23456789 +20
7045.30
120 dB
100 dB
80 dB
60 dB
40 dB
20 dB
0 dB
7035
7040
7045
7050
7055
7060
7065
7070
7075
LSB
7055.98 KHz
Step 12 Hz
Peak Input L: -38 dBFS R: -37 dBFS


Rocky 1.5
File View Tools Help
1.8
23456789 +20
7045.30
7060
7055
7050
7040
7035
Start
Rocky

Two features of the ***Rocky*** software to note are:

i) Spectrum display, clever processing

ii) Automatically optimises ***I & Q*** phase and amplitude for best image rejection

WinRad allows the phase and amplitude to be manually adjusted.

I have been unable to run the ***M0KDK KDKSRD*** program. This does not recognise my soundcard, and also throws up a number of errors.

The ***Flex*** software for the ***SDR-1000*** has been modified to run with the ***SoftRock*** receiver, however I have not been able to install it. Requires a number of ***Microsoft*** updates (MDAC) etc, good old ***Microsoft*** does not recognise my genuine copy of ***Win98se***, guess again expecting ***XP***.

There is a version the ***SM5BSZ Linrad Linux*** program that has been converted to use the soundcard as an input, and compiled as a standalone executable file which runs under ***Windows***.

Having purchased a 24 bit / 96kHz sampling soundcard to play with, I discovered a major problem, ***Win98se*** ***does not*** support 24 bit sound, and no drivers are available.

24 bit sound operation requires ***Win2000*** or ***XP***.

Anyway, it is possible to demonstrate the ***SoftRock*** receivers using the programs we have got working.

I've taken the precaution of making some recordings of 40m ***I/Q*** signals in case of problems with the proximity of the aerial etc.

So without further delay ……

We will try and answer any questions as we go along.


What did you do at the Weekend?