

## A network analyzer (150MHz)

To measure the frequency response of a filter (or an amplifier), we can use a sine wave generator and an oscilloscope.

At each frequency, we have to measure the frequency, the amplitude at the input, the amplitude at the output and also the phase.

A lot of time is necessary to do that, and the measurement precision is not very good.

It is also difficult to measure a capacitor, an inductor or a resistor at several frequencies.

One solution is to use a **network analyzer**.

The network analyzer we will describe here, is mainly a sine wave generator driven by a personal computer (PC) and a voltmeter connected to the same PC.

The voltmeter is like a synchronous detector, it measures the amplitude and the phase of the signal at the frequency of the generator.

The frequency range is from 1000Hz to 150MHz, the frequency resolution is approximately 0.093 Hz.

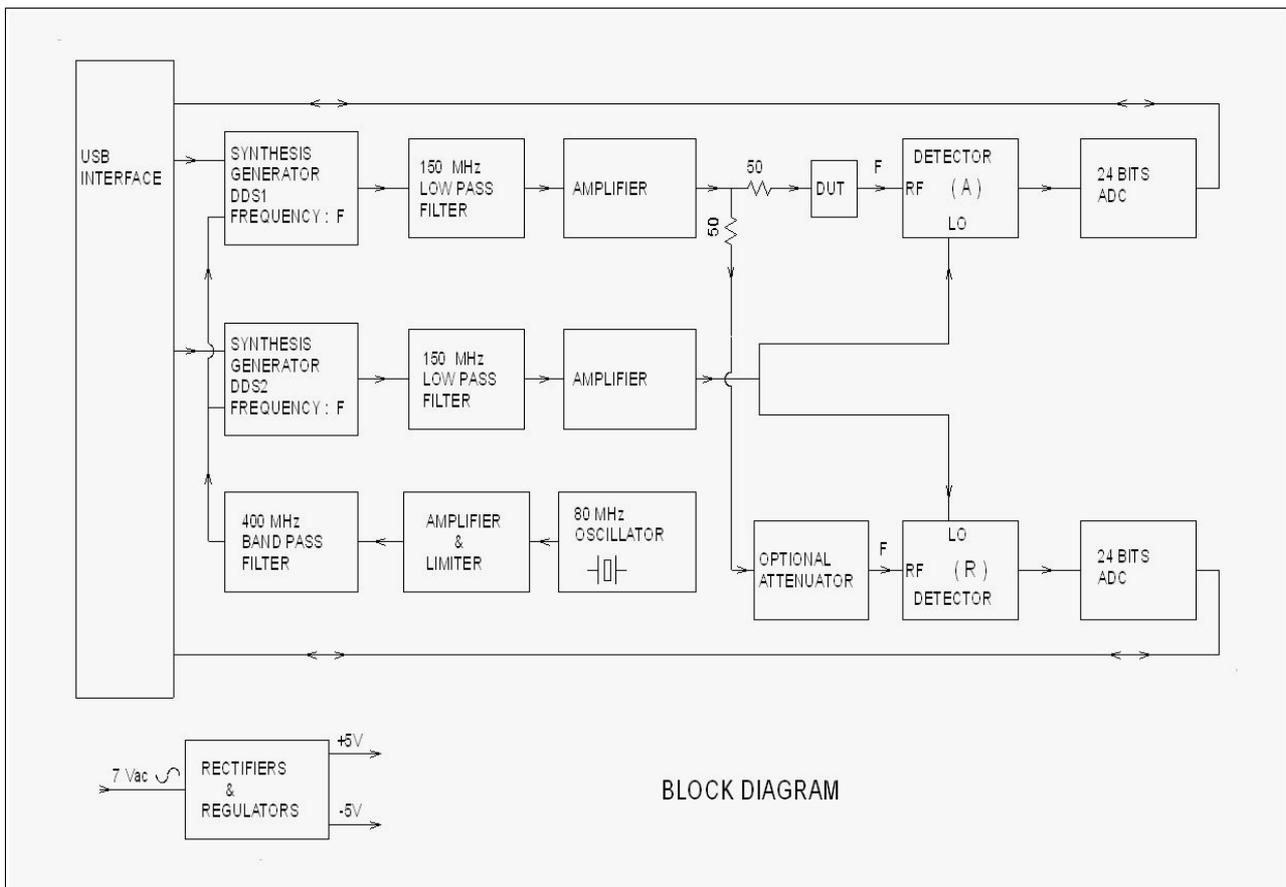
With a slow scanning rate, it is possible to measure from 20 Hz.

The network analyzer measures the **gain and the phase** (or the group delay) of filters and amplifiers (versus frequency). The gain scale is linear or logarithmic (dB scale); the frequency sweep is linear or logarithmic.

A network analyzer which measures at the same time the amplitude and the phase of the gain, is sometimes called: Vector Network Analyzer (VNA).

The network analyzer measures **impedances**, it can display :

- the module and the argument (phase) of the impedance .
- the real part and the imaginary part of the impedance.
- the module and the phase of "S11"
- for a capacitor, the capacitance of the capacitor and the quality factor (Q) .
- for an inductor, the inductance of the inductor and the quality factor (Q) .



## How it works :

Two direct digital synthesis generators (DDS) , (Analog Devices AD9951), are driven by an USB port of a PC.

The two generators are at the same frequency  $F$ , but the phase difference between the two generators is programmable.

DDS1 drives the input of the device under test (DUT); the output of the DUT is the RF input of a detector (Motorola MC1496). This RF input is the (A) input of the network analyzer.

DDS1 drives also the RF input of a second detector (through an optional attenuator). This input is the (R) input of the network analyzer.

The second generator DDS2, drives the local oscillator (LO) inputs of the two detectors.

At the output of the two detectors, there is a low-pass filter.

Two 24 bits analog to digital converters (ADC) , measure the signal at the outputs of the two low pass filters.

These measurements are made for various values of the phase difference between DDS1 and DDS2.

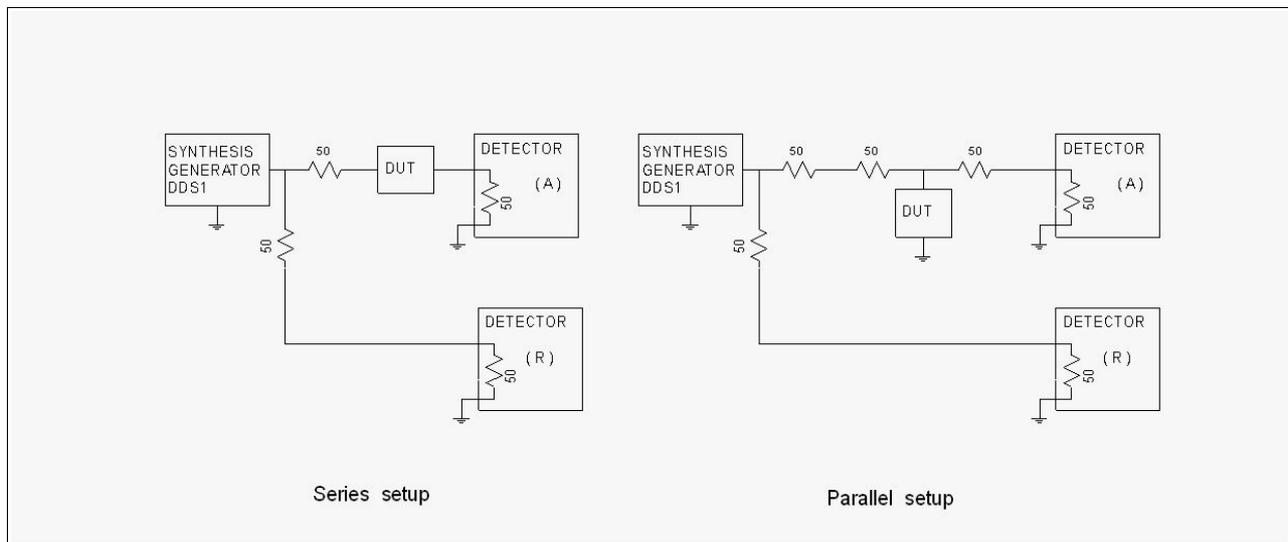
Then, with the comparison of the two signals at the outputs of the low pass filters, the PC can calculate the attenuation (or the amplification) through the DUT; and also the phase rotation through the DUT.

The generators need a 400MHz clock. To get this 400MHz clock, we use a 80MHz quartz crystal oscillator. The fifth harmonic (400MHz) is selected with a 400MHz band-pass filter.

We can also choose to use the PLL of the AD9951; in that case, we only need a 20MHz quartz crystal oscillator.

An AC adapter (7Vac) is used to make the +5V and the -5V needed by the electronic circuits.

To measure **R, L, C** components, we have the choice between the "series" setup and the "parallel" setup.



We can put the DUT in **series** between the generator (DDS1) and the input of the detector (A).

The impedance to be measured modifies the amplitude and the phase of the signal on the input of the (A) detector compared to the amplitude and the phase of the signal on the input of the (R) detector .

If we compare the signal at the output of the (A) detector with the signal at the output of the (R) detector, we can calculate the DUT impedance.

With the series setup, the measurement precision decreases when the impedance of the DUT becomes very low (compared with 50 ohm).

With the series setup, we can measure impedances higher than 1000M $\Omega$ .

We can put the DUT in **parallel** with the detector (A) and the generator (DDS1).

Two 50 $\Omega$  resistors are added (next to the DUT) (these resistors are optional).

If we compare the signal at the input of the (A) detector with the signal at the input of the (R) detector, we can calculate the DUT impedance.

To measure very low impedance values ( compared with 50 ohms), the precision is better with the parallel setup than with the series setup.

Parallel setup is the only solution to measure an impedance with a ground connection : (antenna, or input/output of an amplifier).

The measurement precision is not good if the impedance of the DUT is very high (compared with 50 ohm).

## Electrical schematics

The network analyzer is built with 7 printed circuit boards.

- a DDS1 generator
- a DDS2 generator
- a detector (A)



Two high precision 50 ohm resistors send the signal to the output (A) and to the output (R).  
Capacitors C1,C2 are optional ; with a second analyzer, we can find the best value for C1 et C2.  
The aim of these capacitors is to reduce the reflexions onto the 50Ω resistors.  
The value of C1 and C2 is probably less than 1pF.

We tried to have about the same length of coaxial cable on the outputs (A) and (R).

Output (A) and output (R) are DC coupled, **do not put DC voltage on the outputs !**

The 3.3 volts necessary to the AD9951, is obtained starting from the +5V with the assistance of a divider made up of two resistors (330Ω and 160Ω).

A voltage regulator LM317 provides the 1.8 volts.

The amplitude of the signal at the output of the generator is programmable in the AD9951.

The main weakness of our generator is the presence of a noise at the frequency of 100 MHz (REFCLK/4). To see this noise, we use a high pass filter and an oscilloscope.

This noise is present, whether we use or not the PLL of the AD9951.

We do not know if this noise is also present on the Analog Devices evaluation board.

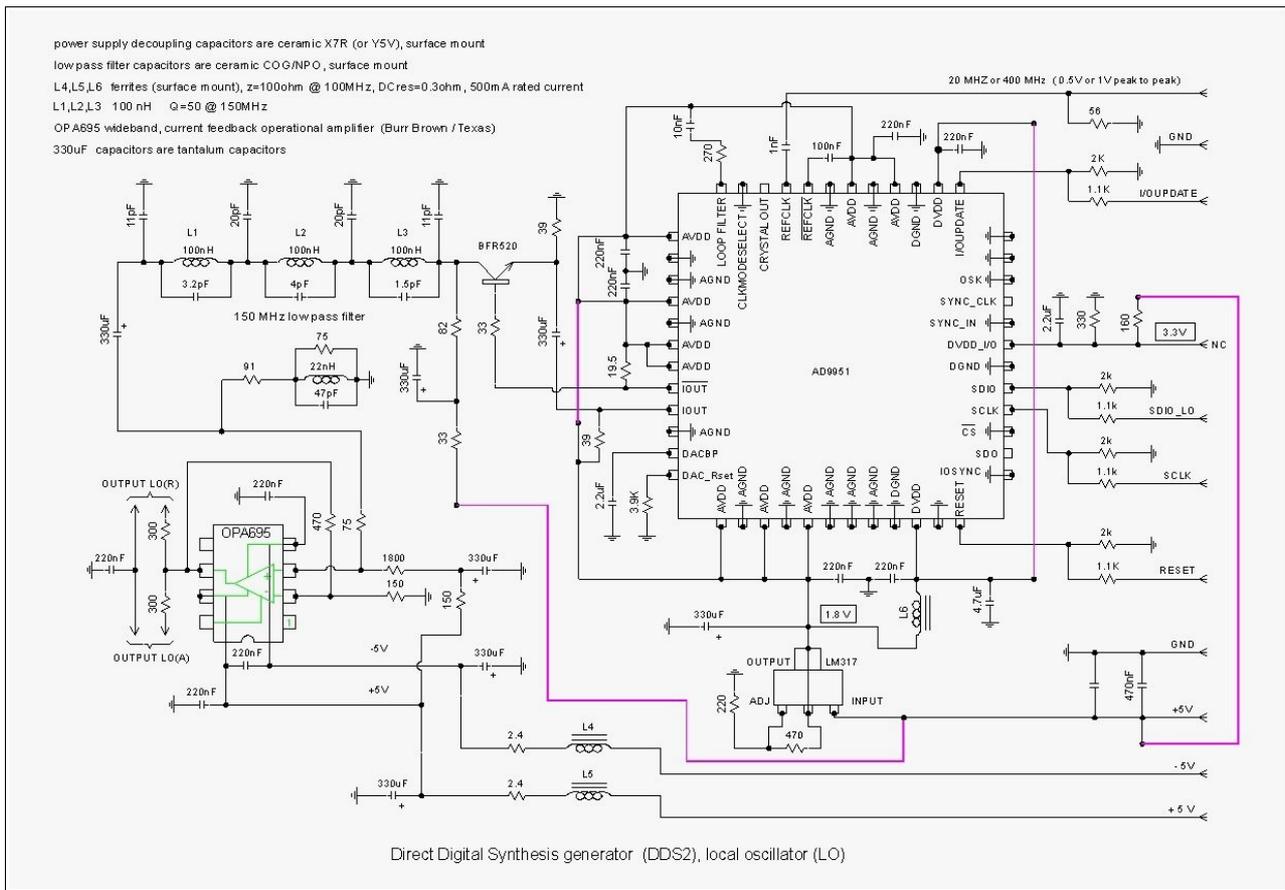
When the amplitude of the signal is 1.61V peak to peak, the amplitude of the noise is 4mV or 5mV peak to peak.

This noise is an amplitude modulated voltage, the frequency is exactly the frequency of the clock divided by four (approximately 100MHz).

A reduction of this noise, would be the welcome!

The principal disadvantage of this noise, is a reduction of dynamics to the value of 105 dB at the frequency of 100MHz +15 Hz and at the frequency of 100MHz - 15Hz (approximately). However, it is rather easy to avoid these frequencies, except if, of course, we wish to measure a component exactly at these frequencies .

# The DDS2 generator



The DDS2 generator and the DDS1 generator are very similar.

The main difference is the presence of a DC voltage (1.67V) on the outputs LO (R) and LO (A).

The output impedance of the generator is 300Ω.

The connection between generator DDS2 and the boards of the detectors uses enamelled wires (very slightly twisted and as short as possible).

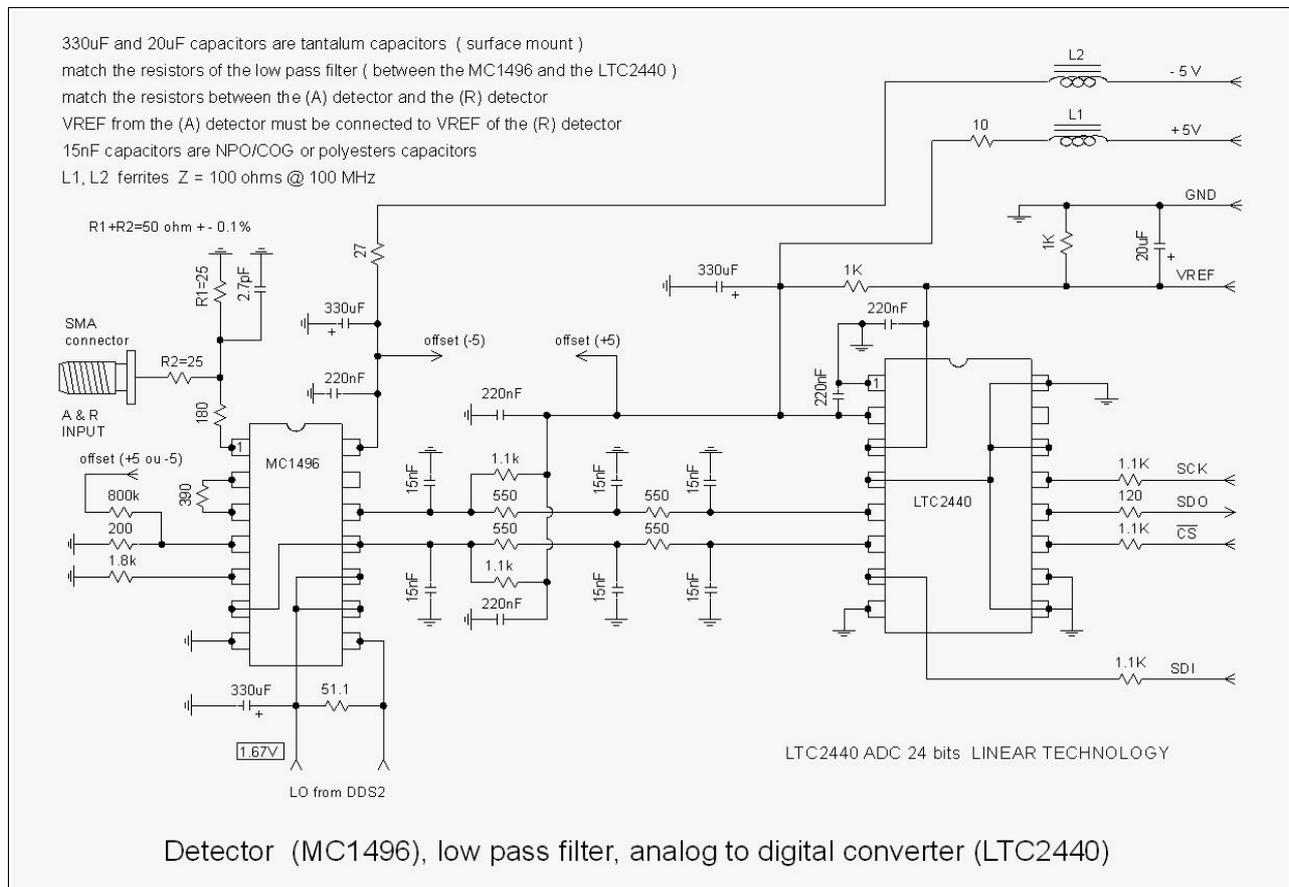
The diameter of the enamelled wires is 0.8mm, it is important that these wires are rather rigid, because the least displacement of these wires creates a large variation of the phase (when the frequency is high).

We tried to have outputs LO (R) and LO (A) symmetrical, mainly the same length of connection.

The polarized capacitors are tantalum capacitors.

You will notice that we used many 330uF tantalum capacitors (as DC block or as power supply decoupling). The main reason is because we have a batch of these capacitors on hand; a lower value of the capacitance is certainly sufficient, especially if you do not need to measure when the frequency is very low.

## The detectors



The detector (A) and the detector (R) are identical.

A balanced modulator (MC1496), mixes the input signal (A or R) with the local oscillator signal. At the output of the modulator, a low pass filter reduces the amplitude of the signal when the frequency is high .

An analog to digital converter (LTC2440) measures the signal.

VREF from the (A) detector is connected to VREF of the (R) detector.

The DC voltage (1.67V) on pins 8, 9 and 10 of MC1496, comes from DDS2 generator .

The (A) and (R) inputs use PCB edge SMA connectors.

There are two resistors at the input (R1 and R2), we must have  $R1+R2=50 \text{ ohm} \pm 0.1\%$

Inputs (A) and (R) are DC coupled, **do not put DC voltage on inputs !**

Do not exceed a maximum signal level of +8dBm on the inputs !

There is no protection on (A) and (R) inputs (to avoid alteration of the input impedance).

The MC1496 is not expensive and easy to replace (in case of failure).

Do not swap the two wires coming from DDS2, pin 8 of MC1496 must be connected to the output of OPA695 through the 300Ω resistor.

On pin 4 (signal input) of the MC1496, there is a resistor (800kΩ) connected to the +5V or the -5V. This resistor is completely optional; its objective is a compensation of the input offset voltage . The

value of the resistor and the voltage to which it must be connected (+5v or -5v), are to be determined when the analyzer is functional.

To find the value of the resistor and where it must be connected, set the frequency of the generator at 200Hz, set the span at 0Hz, start the frequency sweep then stop it. Let the inputs (A) or (R) open. With an oscilloscope, look at the voltage at the output of MC1496. Adjust the value of the resistor to minimize the AC voltage. The objective is to have the best rejection of the local oscillator voltage.

This adjustment of the offset voltage improves ( slightly ) the dynamics when the frequency is less than 1000Hz.

The gain of the MC1496 increases above 100MHz (approximately), then decreases above 150MHz (approximately). To keep the frequency response flat, we added a resistor (180 $\Omega$ ) on the entry of the MC1496 (pin 1).

The detector (A) and the detector (R) must be identical (the most possible ).

To do that, we match the resistors between the detector (A) and the detector (R). The resistors concerned are those close to the MC1496 and the resistors of the low pass filter (between the MC1496 and the LTC2440 ).

The low-pass filter (which makes the connection between the MC1496 and the LTC2440) is differential, we matched the 1.1K resistors, we matched the 550 $\Omega$  resistors (to be the most symmetrical).

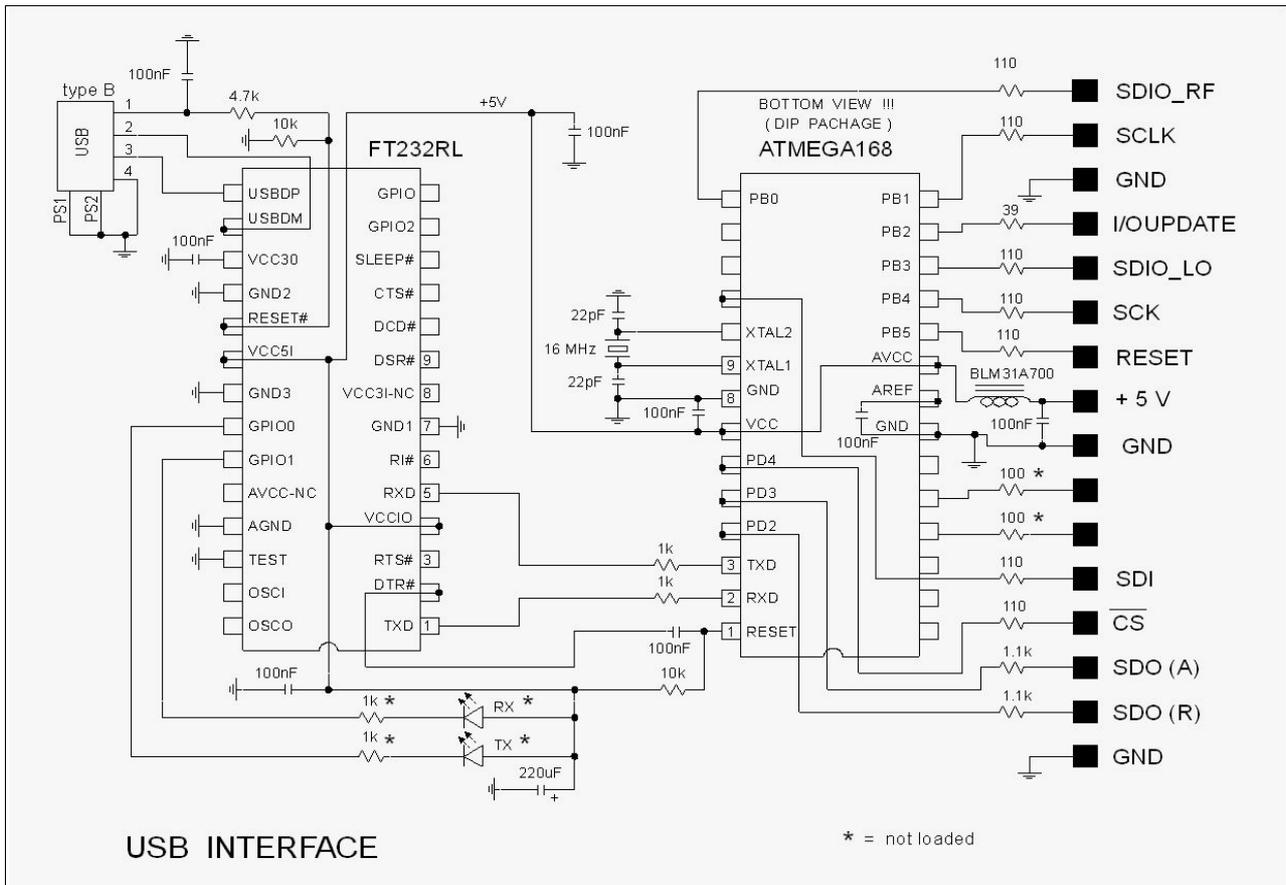
Check (with a DC voltmeter) the outputs of the MC1496. They must be practically at the same potential when the outputs (A) or (R) are open.

We tried to replace the MC1496 with an HFA3101 . The behavior of this IC is better when the frequency is high.

At the low frequencies, the phase is less constant with the HFA3101 than with the MC1496; there is a phase rotation , approximately 20 milli-degrees (corrected by normalization), but we did not yet understand the root cause.

The MC1496 being less expensive and easier to buy, we chose to keep the MC1496 in our design. But if (in the future), we want to measure above 150MHz , we will have to use an other modulator than the MC1496.

## The USB interface



The USB interface uses the work of the Arduino project <http://www.arduino.cc/>

The electrical schematic is an approximative copy of the Diecimila board (from the Arduino project).

An FT232RL (manufacturer FTDI) changes the USB signals to communicate with the serial port of an ATMEGA168 micro-controller (from ATMEL).

For the software, this interface is seen as a virtual COM port (57600 bits/s , 8 bits, 1 stop bit, no parity).

To program the micro-controller there are two possibilities :

1 - Buy a Diecimila board, and load the network analyzer program with the Arduino software (through the USB port).

2 – Buy an ATMEGA168 and program the flash memory with a parallel port programmer. The Arduino project has a parallel port programmer, but we do not succeed with this programmer ? We have used a parallel port dongle DT006 with the PonyProg software .

<http://www.lancos.com/prog.html>

The DT006 dongle is very easy to built : 4 resistors and a parallel port connector.

The PonyProg software needs to access directly the parallel port, it is not possible with windows 2000 or xp; it is necessary to install a software like «userport» or «Porttalk» to allow this direct access.

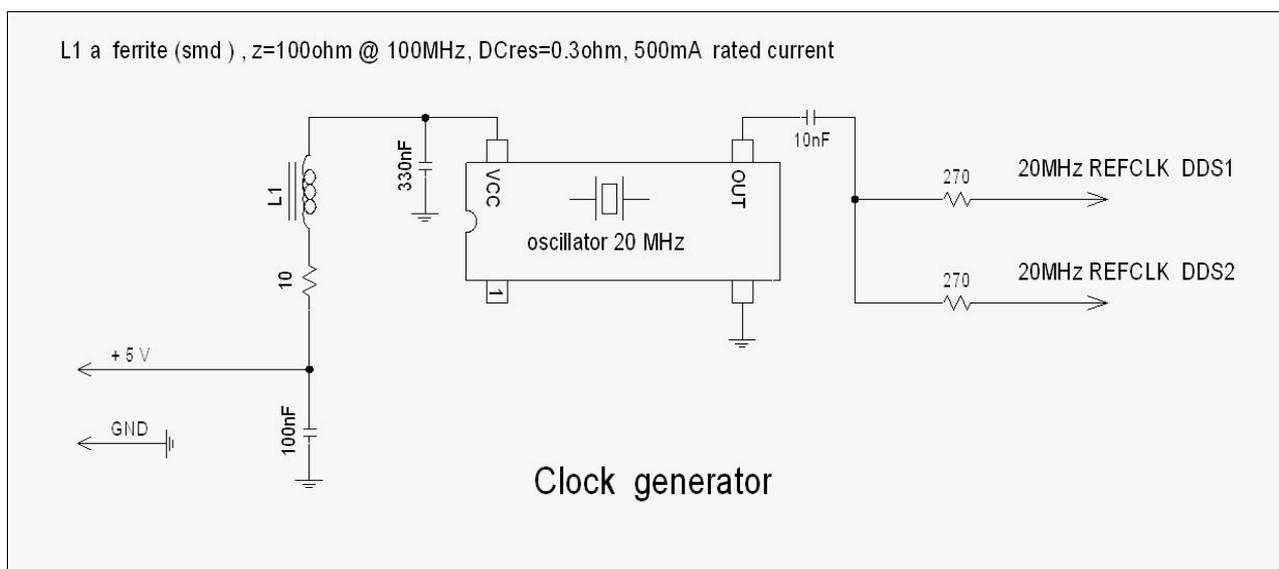
The power supply for the USB interface comes from the network analyzer ; the + 5volt USB supply is only used to reset the FT232RL.

The micro-controller program is small.

On our USB transfers there is no error detection.

The printed circuit is the same for the (150MHz) analyzer than for the (60MHz) analyzer. We only added a wire to connect the signal SDI to the pin 11 of the ATMEGA168.

## The clock generator at 20MHz



DDS generators (AD9951) need a clock at the frequency of 20 MHz (or 400MHz).

To get the 20MHz clock, we use a 20MHz quartz crystal oscillator. At the output of this oscillator, the signal is square, the amplitude is 5V.

The 270Ω resistors and the 56Ω resistors (on DDS boards) lower the amplitude of the signal at a level compatible with the AD9951.

A coaxial cable is used to connect this clock to the DDS generators. The characteristic impedance of this cable is 50Ω or 75Ω.

The main drawback of this 20MHz clock (compared to the 400MHz clock) is a reduction in dynamics at the frequencies of 20MHz +15Hz and 20MHz - 15Hz ; dynamics is reduced to 115dB at these frequencies. We also found a small reduction in dynamics on the third harmonic.

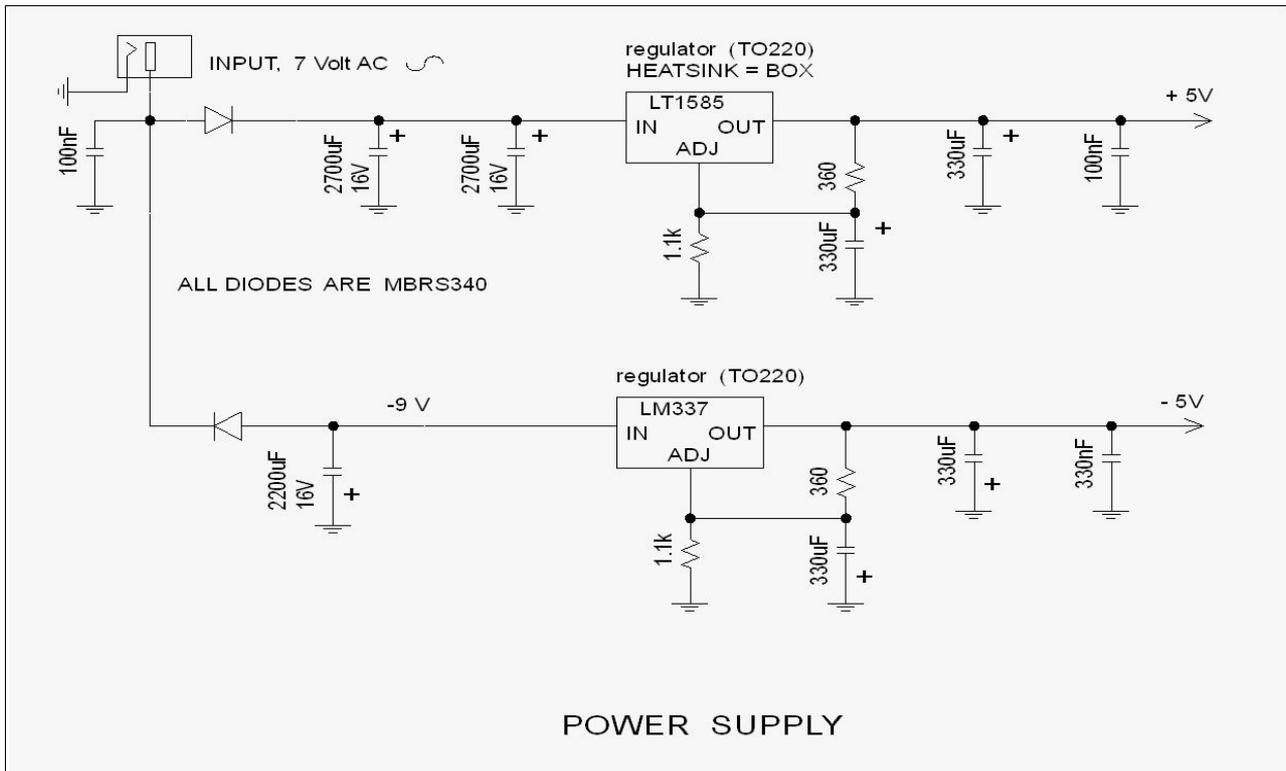
## The clock generator at 400MHz

To see the description of this clock :

<http://golac.fr/horloge-400MHz-pour-l-AD9951>

Description is in French; we will translate it into English soon.

## Power supply



The power supply uses a small external 7 volts (rms) AC adapter (7 volts measured without load). The diodes are 3 Amperes schottky rectifiers (surface mount). The +5 volts regulator (LT1585) uses the box as heat-sink (it is necessary to have an electrical insulation between the regulator and the box). Resistors, diodes and capacitors are surface mounted components, excepted the 2700uF capacitors. 330uF capacitors are tantalum capacitors, a smaller capacity is acceptable. The -5V comes from a LM337 voltage regulator (without heatsink).

## The software

Do not expect a beautiful program , the capabilities of the programmer are very limited ! All the programs are written in basic language (freebasic). <http://www.freebasic.net/>  
A linux version and a windows version are available.  
To write the windows version, we use Fbide (as development interface) <http://fbide.freebasic.net/>  
To write the linux version, we use Geany <http://www.geany.org/>

There are four programs :

Zmeasure.exe : to measure the impedances of the resistors, capacitors ..etc

Fmeasure.exe : to measure the insertion gain of the filters or amplifiers.

Zdisplay.exe : to display later on the measurements made with the Zmeasure program

Fdisplay.exe : to display later on the measurements made with the Fdisplay program.

**Video mode :** The programs use a windows of 1024\*768 pixels.

It is not possible to resize the windows.

If the PC has a 1024\*768 screen, it is better to use the full screen mode (with alt + enter).

If the screen resolution is higher than 1024\*768, it is better to display this application in a window of 1024\*768 (the picture is sharper).

## **USB FT232R driver installation**

**windows :** install the FT232 driver, choose the Virtual Com port (VCP) driver

<http://www.ftdichip.com/Drivers/VCP.htm>

Connect the USB cable and the power supply of the analyzer.

In the windows menus, find the COM port address of the FT232R USB interface (COM3 or COM4 ..etc) (start > run > devmgmt.msc )

Configure the COM port : 57600 bits/s , 8 bits, 1 stop bit, no parity .

Run Fmeasure.exe and press the F2 key to write the COM port address.

**Linux :** (with the UBUNTU distribution)

We do not need to install a driver; but it is necessary to remove a braille program : brltty

Type the command : sudo apt-get remove brltty , or with synaptic remove the brltty program.

Connect the USB cable and the power supply of the analyzer.

In the directory /dev/ there is a file named ttyUSB0 or ttyUSB1 or ttyUSB2 ...etc

Run Fmeasure.exe and press the F2 key to write the named of this file (ttyUSB0 or ttyUSB1 or ...).

To get more information : <http://www.arduino.cc/>

The name of the board is diecimila.

The first time we run Fmeasure or Zmeasure , a small file named CONFIGU is automatically created.

This file contains :

- the COM port address (windows) or the name of a file (ttyUSB0 or ttyUSB1 or ...) (linux)
- the value of the reference resistor used for the calibration
- the value of the parasitic series inductance of the reference load
- the value of the parasitic shunt capacitance of the reference load
- the value of the amplitude and phase mismatch between the input (A) and the input (R).
- the exact value of the clock frequency (about 400 MHz or 20MHz)

The CONFIGU file is not a text file, you can write in this file through Fmeasure or Zmeasure.

## **A (small) Help** for Fmeasure and Zmeasure

- To start or stop a scan, press the space bar or use the «start/stop» button; when the scan is stopped, you can access the menu.
- To move across the menu, use the mouse .
- To change a number, put the mouse cursor on this number then type the new number + “enter”
- To increment/decrement a digit of a number, put the mouse cursor on the digit, then move the mouse scroll wheel.

- To move the markers use the mouse.
- The frequency sweep range can be set with "**freq max**" and "**freq min**" or with "**center**" and "**span**".
- To copy **marker1** to Freq Min and **marker2** to Freq Max, type CTRL+V (it's like a zoom function, but you have to scan again). If marker1=marker2 then CTRL+V copies marker1 to Freq Center.
- When a scan is finished, we can stop the scan with the space bar or the stop button.  
After that, choose the parameter to be measured: gain, impedance, S11 etc; adjust the scales to your needs.  
If the frequency scale is modified, press the space bar to start a new scan.
- **To setup the scales**, we can use: k = kHz, m = MHz, k = kΩ, m = MΩ, u = μ, n = nano, m = milli
- **The number of measurements per scan** «meas/scan» can be set at the following values: 51, 101, 201, 401, 801. The default value is 401.  
When you change the «meas/scan» value, the calibration (normalization) is cleared and the measurement memory is cleared.  
On the graph, the measurement points are connected by the mean of straight lines.  
The markers indicate the measurement values of the nearest measurement point (There is no interpolation).
- **Number of phases per cycle**: «phases/cycle»  
Some menus of this analyzer, usually do not exist on commercial analyzers; it is necessary to give some explanations about the principle of the measurement made by this analyzer.  
When we make a measurement, the two generators, DDS1 and DDS2 have the same frequency and the same phase, we make the first measurement. Then, we shift the phase of DDS2, the phase shift is 90°; we make the second measurement. Then, we do the same for 180° and 270°. We have got 4 points of measurements distributed over 360° (one cycle). We say that we have 4 phases per cycle.  
This number of phases per cycle is adjustable, we have the choice between 4, 8, 16 or 32.  
The scanning rate decreases when the number of phases per cycle increases.  
The interest to increase the number of phase per cycle is to decrease the sensitivity of the detectors to the harmonics present on the inputs (A) and (R) (or generated by the detectors).  
If the number of phase per cycle is 4, the amplitude of the third harmonic is measured with an amplitude divided by 3 (compared to the fundamental frequency), the fifth harmonic is divided by 5 and so on for the other odd harmonics.  
If the number of phase per cycle is 8, the third and fifth harmonics are completely removed, the seventh and the ninth harmonics stay but are divided by 7 and 9.  
If the number of phase per cycle is 16, the harmonics 3, 5, 7, 9, 11 and 13 are removed, the fifteenth and seventeenth harmonics stay (divided by 15 and 17).  
If the number of phase per cycle is 32, the harmonics up to 29 are removed.  
In the case of our analyzer, the value of 8 phases per cycle seems to be a good compromise, with 4 phases per cycle, often we lose precision in measurements (4 is not recommended).  
If we suspect an harmonics measurement problem, (mainly for the measurement of a band pass filter with a strong attenuation and with stiff sides), we can make a checking with 16 or 32 phases by cycles.  
Note: the analyzer 60MHz (that we built previously) operates in a different way. Harmonics 3 and 5 are removed by the principle of measurement (7 measurement points per cycle), the higher harmonics are removed by the filter of the analog to digital converter (ADS1251).
- **Number of cycles per measurement**: «cycle/meas»

We saw in the preceding paragraph, that, for each analyzed frequency, the number of measurements is equal to the number of phases per cycle: 4, 8, 16 or 32.

It is possible to repeat this cycle of measurements several times, while always remaining at the same frequency.

When we make the analysis of a signal by Fourier transform; the longer the analyzed sequence, the better is the frequency resolution.

In the case of our analyzer, every signal (on the inputs (A) and (R)) which is not exactly at the frequency of DDS2, will have a null value, if the duration of the analysis becomes increasingly long.

When we increase the number of cycles per measurement, we reduce the bandwidth of the measuring filter, then the noise is reduced and the dynamic is improved. Number of cycles per measurement is equivalent to : resolution bandwidth.

The number of cycles per measurement is adjustable; we have the choice between 1, 2, 4, 8, 16, 32, 64, 128, 512, 1024 or 2048.

The main drawback : the sweep time is proportional to the number of cycles per measurement.

- **ADC Over Sample Ratio (OSR) :**

The analog to digital converter (LTC2440) uses a clock to carry out its operations of conversion. The number of clock cycles necessary to carry out a conversion is programmable. The ADC (LTC2440) is like a low-pass filter, when we increase the number of clock cycles per conversion ; we decrease the cut-off frequency of the filter (and in the same time we decrease the measurement noise).

This number of clock cycles per conversion is directly connected to a parameter named : Over Sample Ratio (OSR). In our application, the OSR can take the values: 256, 512, 1024, 2048, 4096, 8192, 16394 or 32768.

The conversion rate and the -3dB bandwidth are inversely proportional to the oversample ratio (OSR).

At the output of the modulators (MC1496) there are many undesirable frequencies, these frequencies disturb the measurement.

The highest frequencies are eliminated by the RC filter (at the output of the MC1496), but, when the frequency of the analysis becomes low ( lower than a few tens of KHz) the RC filter becomes completely ineffective, then the filter of the ADC does the job.

When we want to measure at increasingly low frequencies, we must increase the OSR, in this way we decrease the cut-off frequency of the ADC. If we do not decrease the OSR some noise appears onto the measurements and the dynamics decreases.

It is useful to look at the LTC2440 specification; ( the specification is more comprehensible than my explanations !).

The following table can help to chose the value of the OSR.

Minimum frequency = 2000Hz	then OSR must be > or = 512
=1000Hz	1024
=500Hz	2048
=250Hz	4096
=125Hz	8192
=61Hz	16394
=20Hz	32768

We will try to add a mode OSR = automatic, in which the OSR will change automatically according to the analyzed frequency.

Note: In the analyzer (60MHz) that we built previously, the frequencies of generators DDS1 and DDS2 were shifted by 25Hz, and the analog to digital conversions were made at regular

intervals with the assistance of an external clock.

In this new analyzer (150MHz), in which we have to change the frequency of the analog to digital conversions (by the mean of the OSR), it is easier to have the two generators at the same frequency and to use a shift of the phase.

- **Stabilization delay (ms)**

When we modify the frequency of generator DSS1 (to measure the next frequency), the amplitude of the currents circulating in the device under test, need a certain time before being stabilized.

To study the frequency response of a band-pass filter which has a band-width of 1Hz, we must expect a stabilization time of the order of one second.

That is the goal of the parameter “stabilization delay”, it is used to add a stabilization delay after each frequency change (to wait the stabilization of the amplitude of the currents in the device under test).

To know if the stabilization time is sufficient, it is necessary to double its value and to see if the frequency response is modified. If the frequency response is modified, that means that the stabilization delay was not sufficient.

- **Amplitude of the signal at the output of the generators : «amplitude (dBm)»**

The amplitude of the signal is adjustable between 8dBm and -40dBm. This adjustment of the amplitude is carried out by programming the AD9951. That is quite practical.

But this adjustment of the amplitude is done to the detriment of the number of bits of the digital to analog converter (DAC). The AD9951 has a 14 bits DAC, each time we divide the amplitude by 2 (that is to say -6dB), the DAC loses one bit.

If we wish to have the cleanest possible signal at the output of the generators we do not recommend to use this adjustment for the strong attenuations. An external attenuator will be preferable in that case.

The signal at the output of the AD9951 is polluted by a noise (at the frequency of 100MHz); this noise decreases a little with the attenuation, but not as much as the signal.

When we decrease the amplitude, the signal to noise ratio is thus degraded (an external attenuator does not have this disadvantage, at least much more slightly).

- **ADC over-range**

When the amplitude of the signal (on the inputs (A) or (R)) becomes too important, the signal at the input of the analog to digital converters, can exceed the maximum limit of the converters (this limit is equal to VREF, that is to say 2.5volts in our case).

To inform the user that the precision on the measurement is not guaranteed, an horizontal line is traced under the curve at the places where the capacity of the ADC is exceeded. At the same time a “warning” is briefly written.

- **Change of the measuring parameters**

It is possible to stop the scanning, then to change certain measuring parameters, and to start again the scanning at the place where it had been stopped.

The parameters that we can change are : “cycle/meas”, “ADC oversampling ratio”, “phases/cycle”, “amplitude”, “stabilization delay” as well as the vertical scales and the data that we have chosen to display.

On the other hand, if we change the other parameters, the curve will be erased and the sweeping will start at the frequency “fmin”. These parameters are: “fmin”, “fmax”, “fcenter”, “span”, “meas/scan”, “normalization”, “calibration”.

The fact of changing the scanning rate according to the analyzed zone, often saves time. We measure with a slow scanning only the difficult places (with strong attenuations or with sharp resonances where it is necessary to await stabilization).

- **Adjustment**

"Adjustment" is used to correct the difference of gain between channel (A) and channel (R). In the main screen of the Fmeasure program, there is a button : "Adjustment" . if we click this button, that opens a new screen; in this screen there are fields in which we can write the ratio "gain(A)/gain(R)" ; as well as the difference of phase between (A) and (R) , (measured at low frequencies, for example at 5kHz). To input these data, follow the indications written onto the screen.

On the same screen, there is a field to input the exact frequency of the clock.

If the frequency is lower or equal to 41MHz, then the PLL is programmed to multiply by 20 the frequency of this clock; in that case a clock close to 20MHz must be used.

If the frequency is higher than 41MHz, then the PLL is not used; in that case a clock close to 400MHz must be used.

Following a change of frequency between 400MHz and 20MHz (or between 20MHz and 400MHz), it is necessary to leave the program, then to start it again ( in order to start (or stop) the PLL of the AD9951).

These parameters (gain(A)/gain(R), phase(A)-phase(R) , clock frequency) will be saved on the hard drive , in a file named "config", (config is not a text file).

Note : the first time we start the analyzer, the «config» file does not exist. By default, the PLL is not used; if we use a clock at 20MHz, the first thing to do , is to input the frequency of the clock to start the PLL of the AD9951.

- **Length :**

It is possible to add an electrical length (positive or negative) in series with the DUT. It is useful only if we want to measure a component without normalization or calibration. "length" allows to compensate the difference of length between the (A) channel and the (R) channel. To find the value of "length", use Fmeasure, replace the DUT by a through connection , when a scan is finished adjust the value of "length" to have the smallest phase (between "fmin" and "fmax") . To do that, put the mouse cursor on a digit of "length" , then use the mouse scroll wheel to adjust the value of "length".

- **Normalize :**

With the Fmeasure program, it is possible to measure two ports devices (filters, amplifier etc) with or without normalization . Normalization improves the precision (normalization corrects the gain differences and the phase differences between the channels (A) and (R), and that for all the frequencies of measurement. ).

To normalize : replace the DUT by a through connection, in the setup choose "normalize : asked" then use the "start normalize" button to start a scan. When the scan is finished, remove the through connection and put the DUT; now the new measurements will be normalized.

Normalization is not calibration, because it does not correct the errors coming from the input and output impedances of the analyzer.

The input and output impedances of this analyzer are close to 50Ω (see the measurements).

With the Fmeasure program we can only normalize, we can not calibrate, but the differences between calibration and normalization will be very low because the input and output impedances are close to 50Ω.

With the Zmeasure program, it is possible to measure (the impedance of two ports devices) with or without normalization, but only if we use the "series setup"

With the "parallel setup", we could have allowed the measurement of the impedances without making calibration; but we have estimated that the precision of the measurement would be penalized. With the "parallel setup", the calibration is obligatory.

- **Calibrate :**

With the Zmeasure program, if we use only the "series set up" , it is possible to measure components without calibration, but calibration improves the precision.

With the "parallel setup" it is necessary to calibrate.

To calibrate , set the "freq max" and the "freq min", select "**calibrate asked**", then use the "start calibrate" button to start a scan, then follow the indications written onto the screen.

One then proposes to replace the device under test, successively by an open circuit then a short-circuit and finally a reference resistor (generally the value of the resistor is close to 50Ω).

To indicate to the analyzer the characteristics of this reference resistor, click the button "reference load". one can then indicate the exact value of the reference resistor, its parallel capacitance and its serial inductance.

The values of these elements are saved on the hard drive, in the file named "config", (config is not a text file).

- **Copy measure to Memory**

When a scan is finished , with the button «Copy measure to Memory», we can put a measurement to a memory . On the screen, we can **display** simultaneously the content of the memory and a new measurement. It is useful to compare two components, or to see the effect of an adjustment.

- **Save measure to File :** when a scan is finished, we can save the measurement in a file.

We can write a bmp file (this is a copy of the analyzer screen window).

We can also write the numerical measurement data in a file with the extension .s1p or .s2p

With these file formats (Touchstone), we can import our measurements into a S-parameter simulation software (Qucs, RFSIM99 or PORTVIEW for example)

With our programs Fdisplay (for the filters) and Zdisplay (for the impedances) , we can display our measurements . The files with the extension s1p and s2p must be in the same directory than Fdisplay and Zdisplay.

To save the measurement of a one port device, we choose the s1p file format. This file is a text file with 3 columns, the first one is the frequency (Hz), the second one is the s11 amplitude (dB), the third one is the s11 phase (deg).

To save the measurement of a two ports device , we choose the s2p file format. This file is a text file with 9 columns, nb 1 is the frequency (Hz), nb 2 is the s11 amplitude (dB), nb 3 is the s11 phase (deg), nb 4 is the s21 amplitude (dB), nb 5 is the s21 phase (deg), nb 6 is the s12 amplitude (dB), nb 7 is the s12 phase (deg), nb 8 is the s22 amplitude (dB), nb 9 is the s22 phase (deg).

This network analyzer can only measure one parameter at a time.

With Fmeasure, we measure s21 (forward insertion gain), and we save the data in a s2p file.

We swap the input and the output of the DUT to measure s12 (reverse insertion gain), and we save the data in the same s2p file.

With Zmeasure, we measure s11 (at the input) and we save the data in the same s2p file; then we measure s22 (at the output) and we save the data always in the same s2p file.

When we measure s11, it is necessary to put a 50 ohms load at the output.

When we measure s22, it is necessary to put a 50 ohms load at the input.

If the DUT has only passive components then  $s_{21}=s_{12}$ .

To put in a same file all the s-parameters of a two ports device, it is necessary to keep the same start/stop frequencies for all the measurements, and to keep the same number of measurements per scan.

- **Bar graph indicator** : on the bottom right of the screen , there is a bar graph indicator.



This indicator shows the signal level on the (A) input and on the (R) input. Do not exceed 8dBm , to have a good dynamic range set the signal level at +8dBm, to have a good precision reduce the signal level at 0 dBm.

With the Fmeasure program, we can display the signal level on the (A) and (R) inputs versus frequency. To do that, in the "**display**" possibilities chose : **dB, A,R**.

The signal on the inputs (A) and (R) is measured in dBm, but we uses the same scale as the gain in dB.

The amplitude at the output of a DDS generator varies according to the formula  $a(f)=\frac{\sin(\pi*f/f_{clock})}{\pi*f/f_{clock}}$  with  $f_{clock}=400\text{MHz}$ . The amplitude decreases when the frequency increases.

This decrease is partly compensated by the resonator (22nh, 47pF, 51 $\Omega$ ) present at the output of the 150MHz low pass filter.

If we choose "fmin" =1MHz and "fmax"=200MHz (and a linear sweep), we can see the effect of the 150MHz low pass filter of the DDS1 generator.

However, above 150MHz, the amplitude of the local oscillator of the modulators also decreases, creating a decrease of the modulators gain. Above 150MHz, the error on measurement of the input level increases rapidly with the frequency.

Because of the principle of the direct synthesis, a signal at (400MHz-f (DDS)) is also present in the signal generated by DDS1 and DDS2. Above 200MHz, the amplitude increases, the signal we get now, is the signal at (400MHz-f (DDS)). When one programs F (DDS) =250MHz, one obtains a signal at: 400-250=150MHz

- **The group delay** : with the Fmeasure program we can measure the group delay versus frequency.

The group delay ( $\tau(\omega)$ ) measures how much the phase is increasing (or decreasing) when the frequency is increasing,  $\tau(\omega)=-\frac{\partial\theta(\omega)}{\partial\omega}$  with  $\omega=2*\pi*f$  ,  $\theta(\omega)$  is the phase.  $\tau(\omega)$  is measured in second.

For filters, it is sometimes useful to have a constant group delay (in the pass band of the filter). To calculate the phase change, it is necessary to select two frequency points (f1 and f2), (f2 at a small distance of f1) and to measure the phase  $\theta_1$  and  $\theta_2$  at f1 and f2.

The group delay is calculated as  $-(\theta_2 - \theta_1)/(2*\pi*(f_2 - f_1))$ .

The distance between f1 and f2 is the **delay aperture**, ( delay aper in the menu of Fmeasure).

The delay aperture can be adjusted between 0.25% and 16% of the full scan (the lowest limit varies with the number of measurement points per scan).

The best is to select the lowest value, but if the trace is too noisy, it is necessary to increase the

value of the delay aperture.

For the cables, which have a frequency response relatively flat, the group delay represents duration that the signal puts to be propagated along the cable; but for a filter this is not true; the group delay can be positive or negative .

Inside the delay aperture, if the phase increase (or decrease) is more than 180 degrees then the measurement is wrong ( it is necessary to reduce the delay aperture or the frequency span).

- **Source and load impedances :**

To define the frequency response of an amplifier or a filter, it is necessary to specify what is the source impedance and what is the load impedance. The network analyzer has a 50 ohm source and load impedance, to work with different values of impedances it is necessary to add resistors in series or in parallel.

An other possibility is to measure the 4 s-parameters of the 2 ports device (s11, s21, s12, s22) with a 50 ohms source and load impedance; save the measurement in a s2p file. Open this file with an S-parameter simulation software (Qucs, RFSIM99 or PORTVIEW), then adjust the source and load impedances to find the best values.

- **Linearity correction of the modulators (MC1496) :**

When the amplitude of the signal on the inputs (A or R) increases, the gain of the modulator decreases.

To see this gain decrease : put a 10dB attenuator on the (R) input and no attenuator on the (A) input; the signal on the (R) input is low, the gain on this input does not change , but on the (A) input we can see that the gain decreases (a little bit) when we increase the amplitude of the signal.

To correct this linearity defect : by calcul, we add a linearity correction . This correction is proportional to the amplitude<sup>2</sup> . Today, we can modify this linearity correction only in the source code.

When the signal at the input is +5dBm, the linearity correction is 0.023dB

When the signal at the input is 0dBm , the linearity correction is 0.007176dB

When the signal at the input is -10dBm, the linearity correction is 0.000718dB

- **Gain and phase correction as a function of the frequency and the amplitude.**

At low frequencies, the linearity correction of the modulators allows an improvement of the precision of measurements.

When the frequency is above 5MHz, the frequency response of the modulators is also function of the amplitude.

To see this dependence, one puts an attenuator of 10dB on the input (R) and none on the input (A).

One normalizes with an amplitude of -10dBm, the frequency going from 5000Hz to 150MHz, (with a log sweep); the gain scale is 0.02dB/div, the phase scale is 0.2 degrees/div.

When the normalization is finished, one increases the amplitude. At the low frequencies nothing changes (thanks to the linearity correction of the modulators). Above 5 MHz, the gain of the modulators decreases when the amplitude increases, in the same way the phase turns when the amplitude increases.

To summarize : the gain of the modulator is a function of the frequency and of the amplitude; the phase rotation in the modulator is a function of the frequency and of the amplitude.

If we want precision at the high frequencies (for example at 100MHz), we must work with an amplitude lower than 0dBm, or even lower than -6dBm.

We see two possible improvements ( to correct this problem at least partially). One possible improvement is to adjust a mathematical model to correct the gain and the phase according to the amplitude and to the frequency; the other is to replace the MC1496 by a faster modulator

( for example the HFA3101).

Today, we chose the mathematical correction, at 100MHz (for example), the improvement is already very interesting.

The mathematical correction looks like this :

$$\text{gain correction} = 1 + k_1 * \text{amplitude}^{2.1} * (\text{frequency}/100000000)^{1.5}$$

$$\text{phase correction} = k_2 * \text{amplitude}^{2.1} * (\text{frequency}/100000000)^{1.4}$$

(k1 and k2 are constante values)

However, above 10MHz, if we need precision and if the noise is low, it is better to keep the amplitude lower than 0dBm (because the mathematical correction will be never perfect).

- **The dynamics range**

If we look at the dynamics of the analyzer, we can see that , above 30 MHz, the dynamics decreases (see the measurements).

The root cause of this problem comes from the local oscillator and the leakage inside the detectors and also the leakage of the box between channel (A) and (R). To remove partly this problem, add an external attenuator on the input (R) (20dB or 30dB), then it is necessary to normalize. This attenuator is necessary only if one needs a high dynamics when the frequency is high.

- **The duration of a sweep**

The duration of a sweep is function of 5 parameters :

- . the number of phase per cycle
- . the ADC Over Sample Ratio (OSR)
- . the number of mesurement points per scan (meas/scan)
- . the number of cycles per measurement (cycles/meas)
- . the stabilization delay

Here some examples measured on the analyzer :

phases/cycle	ADC OSR	meas/scan	cycles/meas	Stabilization delay	Sweep time
8	512	401	1	1 ms	19,2 s
		201			9,6 s
		101			4,85 s
		51			2,5 s
8	512	401	2	1 ms	37,5 s
			4		64 s
			8		122 s
8	1024	401	1	1 ms	26 s
	2048				45 s
	4096				70 s
4	512	401	1	1 ms	12,8 s
		201			6,4 s
		101			3,2 s
		51			1,6 s

For the moment, one does not have yet established a formula to estimate the duration of a sweeping (with the 5 parameters).

Attention, when one uses only 4 phases per cycle; harmonics 3 and 5 are not rejected, the errors on measurement will be big if the frequency is high and if the input signal is high. If one needs precision, it is better to use 8 phases per cycle.

## **The printed circuit boards**

All the PC boards have the traces on the top, and a ground plane on the bottom ; except the power supply board which is a single face PC board.

The thickness of the PC boards is 0.8 mm for the DDS1 and DDS2 generators, but it is possible to use 1.6 mm PC boards .

The thickness of all the other boards is 1.6 mm ; the material is FR4.

The connections from the traces (on the top) to the ground plane (on the bottom) are made with some small wires soldered (diameter 0.6mm)

To make all the PC boards we used the toner transfer method. We need a laser printer, a copper clad board, a clothes iron and some Epson glossy photo paper for inkjet printers . If we have some defects : it is useful to use a permanent marker to go over any problem areas

The etchant is a ferric chloride solution. During the etching process we pass a brush on the board to speed up the process and to make uniform the etching. We use acetone to remove the toner (while rubbing with a rag) .

Most of the time, the traces are 0.635mm wide (or more), and there is no problem with the "iron method" ; around the AD9951 , the wide of the traces is 0.25mm, be careful.

To help, the layout and the electrical schematic are very similar.

To solder the AD9951 : very often we put too much solder and the solder creates short circuits between the pins. We remove the short circuits with a de-soldering braid. Note that the exposed paddle on the bottom of the package forms an electrical connection for the DAC and must be attached to analog ground. To solder the paddle, we make a small squared hole in the PC board, then, with a small sheet of copper, we solder the paddle to the ground plane.

On the USB interface, the micro-controller, the quartz crystal and the USB connector are loaded on the ground plane side.

To isolate the pins of these components (from the ground plane), we use a milling bit, the shape of the milling bit is a ball.

All the PC board ground planes are electrically connected to the metallic box with the fixing screws.

## **The measurements :**

The first time that we run the analyzer , connect the output (A) to input (A), and the output (R) to input (R) , (with short cables).

Run the Fmeasure software.

With the space bar (or the stop button), stop the scanning. The first thing to do, is the "adjustment" to indicate the frequency of the clock (see the description of the "adjustment" function (in the

help) ).

Then, choose to measure : gain (U) vs frequency ; scan from 2kHz to 150MHz

The gain must be nearly constant and probably between 0.9 and 1.1

If we disconnect the cable on the (A) input , the gain must be zero.

Now, to correct the bad symmetry between the (A) channel and the (R) channel, press the "F5" key (or clic the "adjustment" button) and follow the indications written onto the screen.

By this way, we correct the lack of symmetry " not frequency dependant " .

The " length " correction allows to compensate the difference of length between the (A) channel and the (R) channel.

The symmetry correction and the length correction are not necessary if we use the normalization (or the calibration).

To check the validity of our measurements, we used 0,1% resistors and 1% capacitors .

When the frequency is higher than 10 MHz , it is more difficult : we have to find the value of the parasitic elements of the reference load (used for the calibration) to get the best correlation with capacitors and inductors available data.

A capacitor 's manufacturer (AVX) has a program (SPIMIC) that gives the quality factor (Q) of surface mount capacitors at several frequencies. Johanson Technology provides a software : MLCSoft.

Some programs calculate the inductance and the quality factor of air coils (at several frequencies).

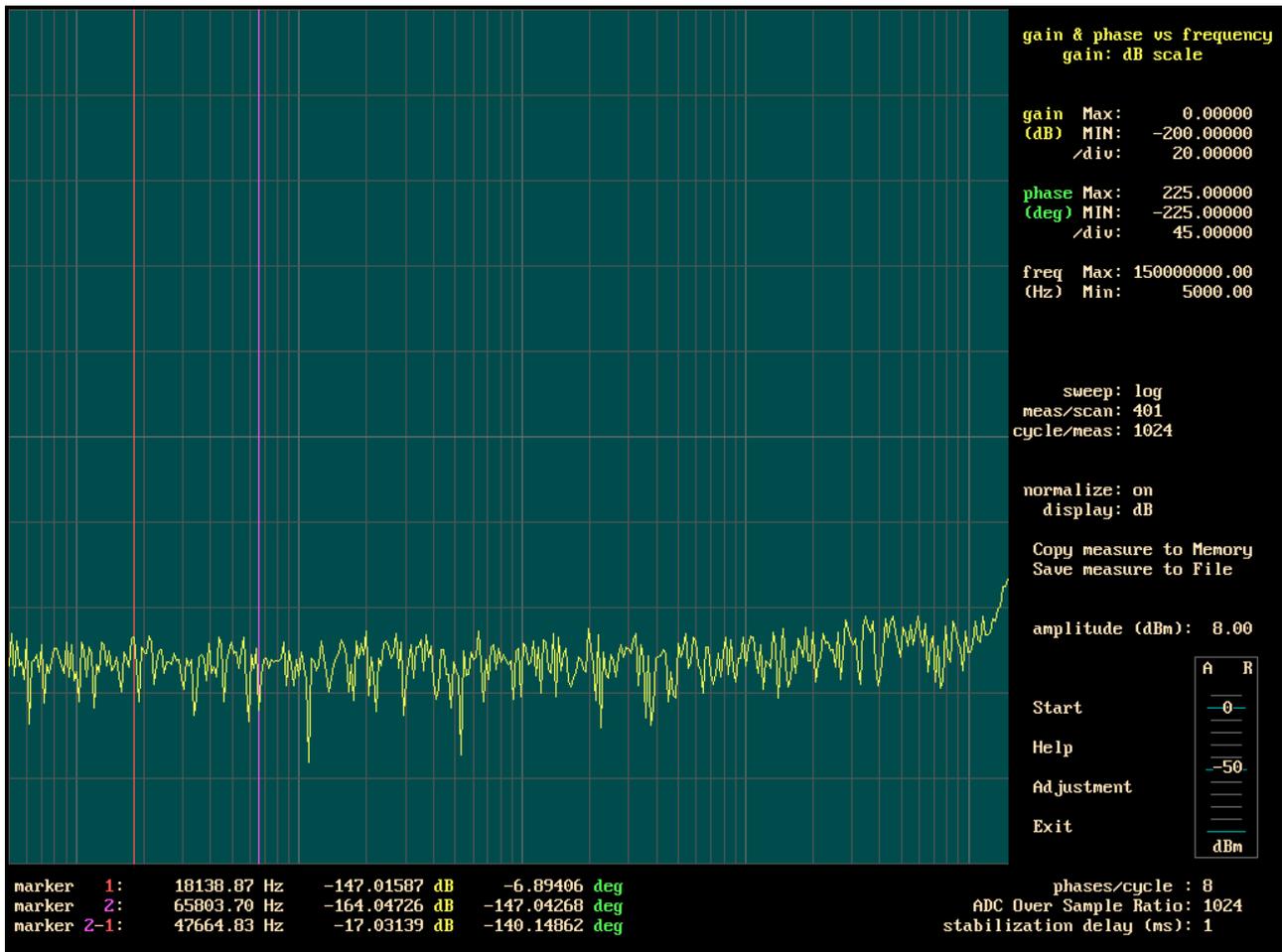
For the capacitors and the inductors, the measurement of the quality factors is always difficult, especially when the frequency is high; because one needs a high degree of accuracy on the measurement of the phase.

In the case of the measurement of a capacitor (for example), to make sure that the measurement is not completely false, it is necessary to associate an inductor (with a very good quality factor) to the capacitor . We measure the quality factor (Q) of the resonant circuit created; this measurement is easy since we need only to measure the frequencies for which the phase rotation is  $45^\circ$ .

We must have  $Q(LC)=Q(L)*Q(C)/(Q(L)+Q(C))$ .

Often, when the frequency is high, we think that we are measuring a capacitor or an inductor; but, in fact we are measuring a more complex component.

For example, when we measure an inductor with the "series setup", we can measure a phase higher than  $90^\circ$  . That does not mean that the quality factor is infinity (or that does not mean that the inductor can give us some energy); it means that our inductor is equivalent to an inductor in series with a delay line.



## Dynamics range of the analyzer

To make this measurement, we put an attenuator (30dB) on the (R) channel, and a through connection on the (A) channel. The amplitude is 8dBm.

We normalize, then we open the (A) channel to measure the dynamics.

The vertical scale is 20dB/division.

The measuring parameters are : OSR=1024, cycles/meas=1024, phases/cycle=8

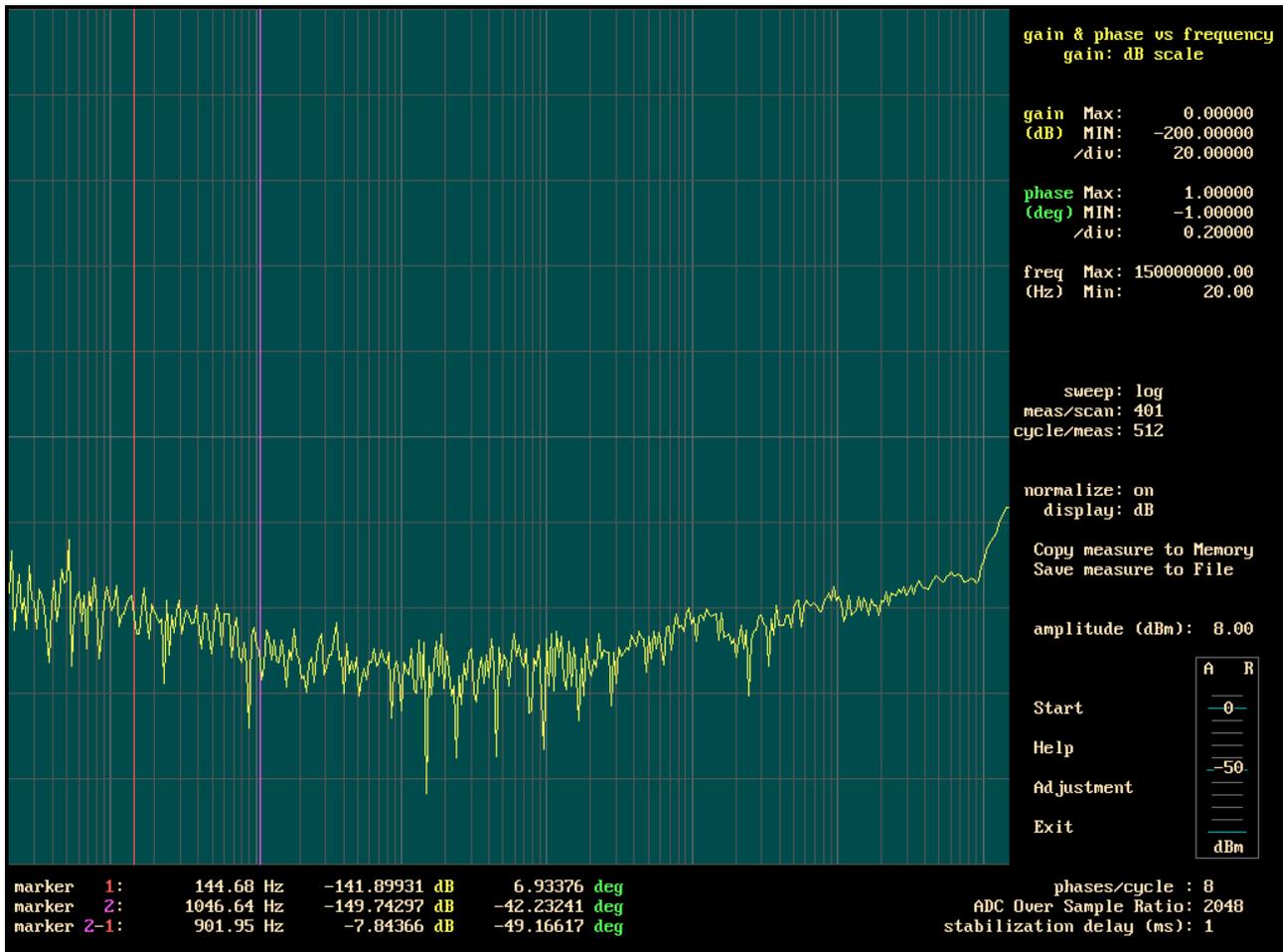
With a so high number of cycles per measurement, it takes about 5 or 6 hours to do this measurement.

Above 130MHz, the leakage between the channel (A) and the channel (R) limits the dynamics.

With a dynamics higher than 140dB, we can measure impedances higher than 1000MΩ when we use the "series setup".

Do not forget that the dynamics is only 105dB at the frequencies : 100MHz+15Hz and 100MHz-15Hz (because of the noise (at 100MHz) of the generators).

If the frequency sweeping is logarithmic, one has little chance to fall on these frequencies; with a linear sweeping that can arrive, one can then indicate that the frequency of the clock is not exactly of 400MHz (by adding a shift of some ppm), or we can change a little bit fmin or fmax.



### Dynamics range of the analyzer (without attenuator on the (R) channel)

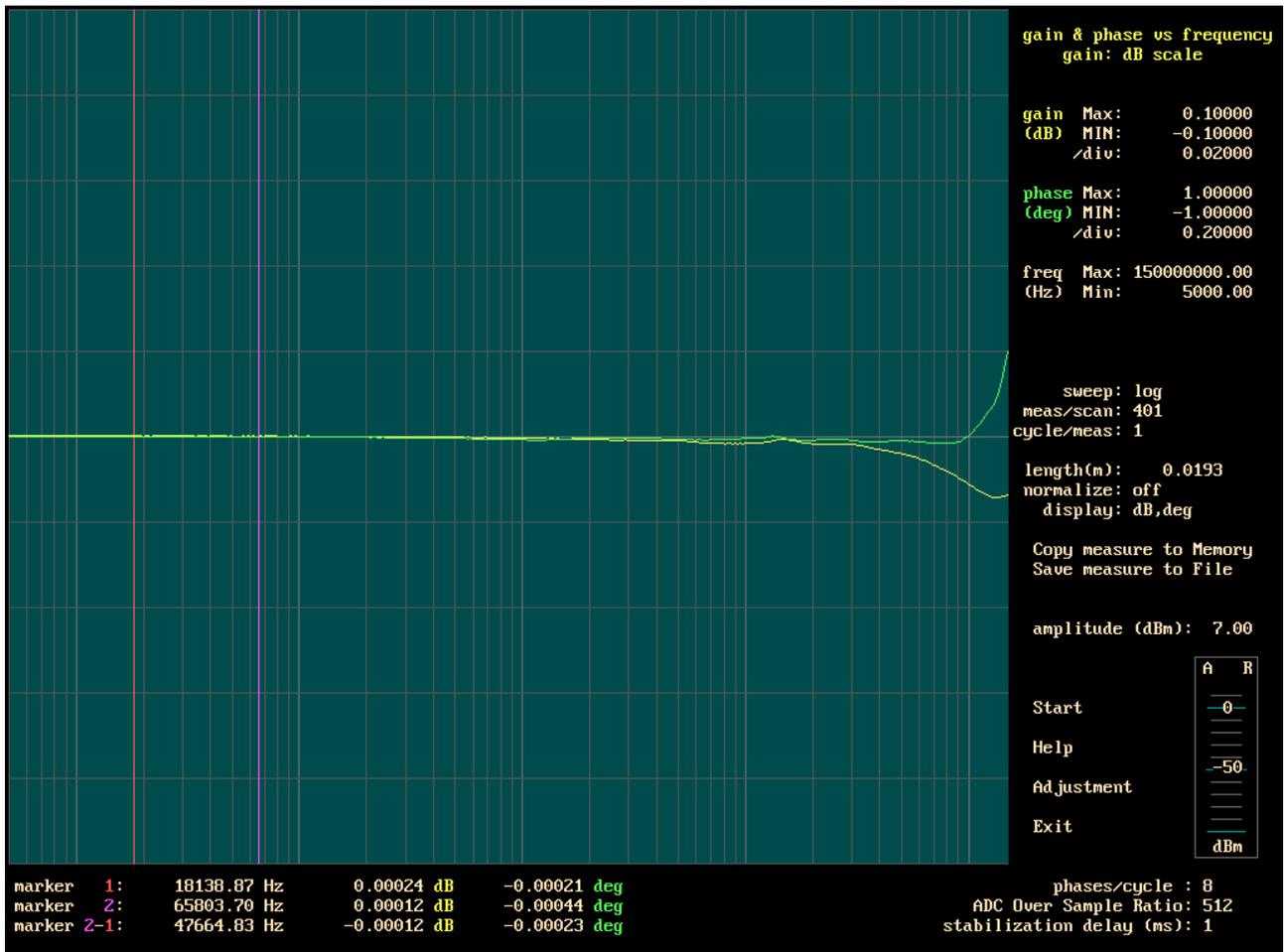
The frequency range is from 20Hz to 150MHz.

Above 2KHz, the measuring parameters are : OSR=1024, phases/cycles=8, cycles/meas=1024, amplitude=8dBm.

At the beginning (20Hz), the OSR = 32768 and the number of cycles/measurement = 32. Then as the frequency increases, we decrease the OSR and we increase the number of cycles/measurement. (we will try to create a mode to do that automatically).

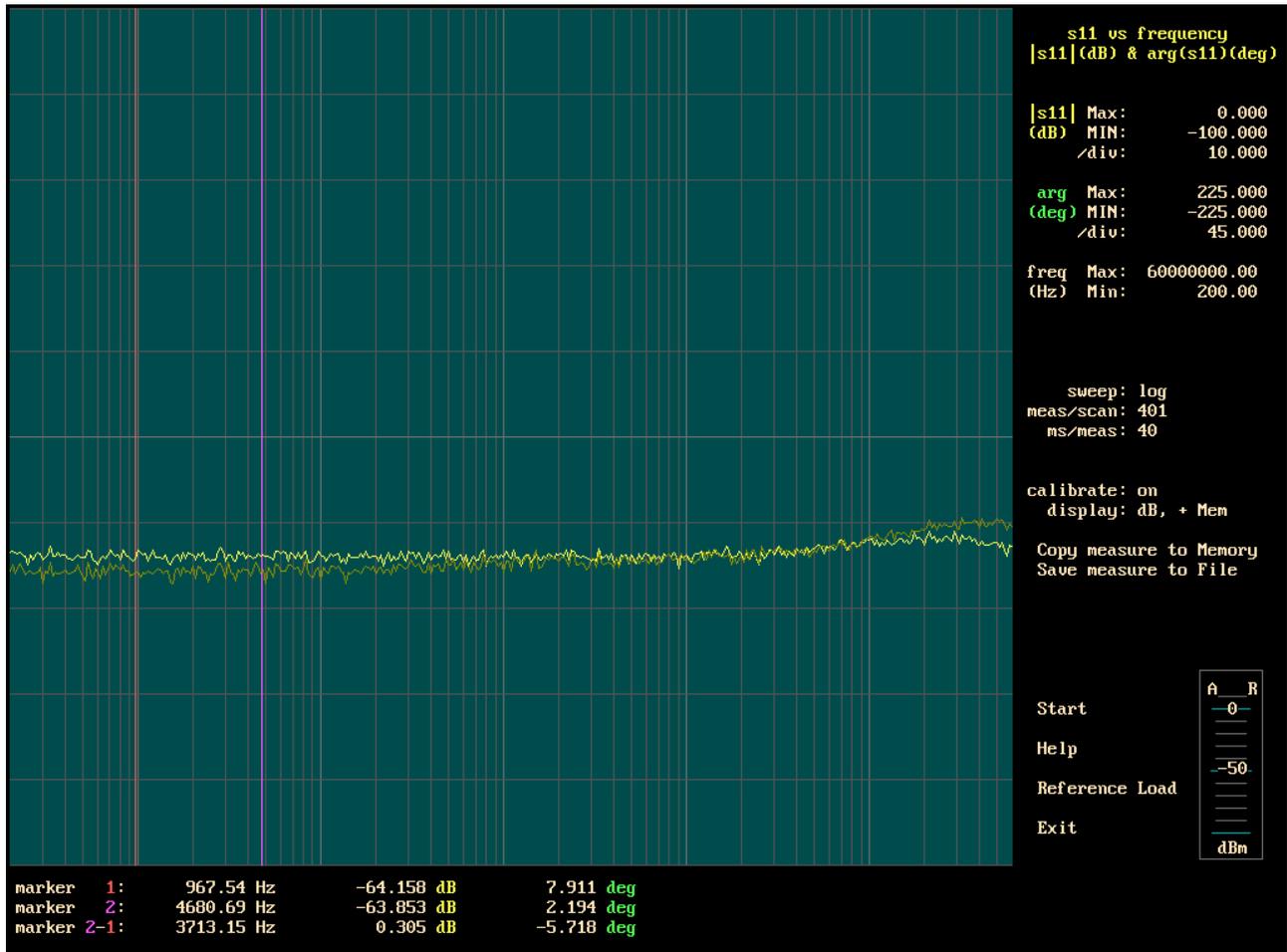
At 150MHz, we notice that we almost lost 20dB on the dynamics compared to the curve measured with the help of an attenuator on the channel (R). When the frequency is high, the lack of separation between the channel (A) and (R) limits the dynamics.

When the frequency is low, a part of the local oscillator signal arrives up to the ADC and limits the dynamics.



On this picture, we can see the **gain differences between the (A) channel and the (R) channel** ; (without normalization, but with the " length correction " and with the " non frequency dependant " correction (adjustment)).

If, in the frequency range of a measurement, these gain differences are acceptable, then it is not necessary to normalize.

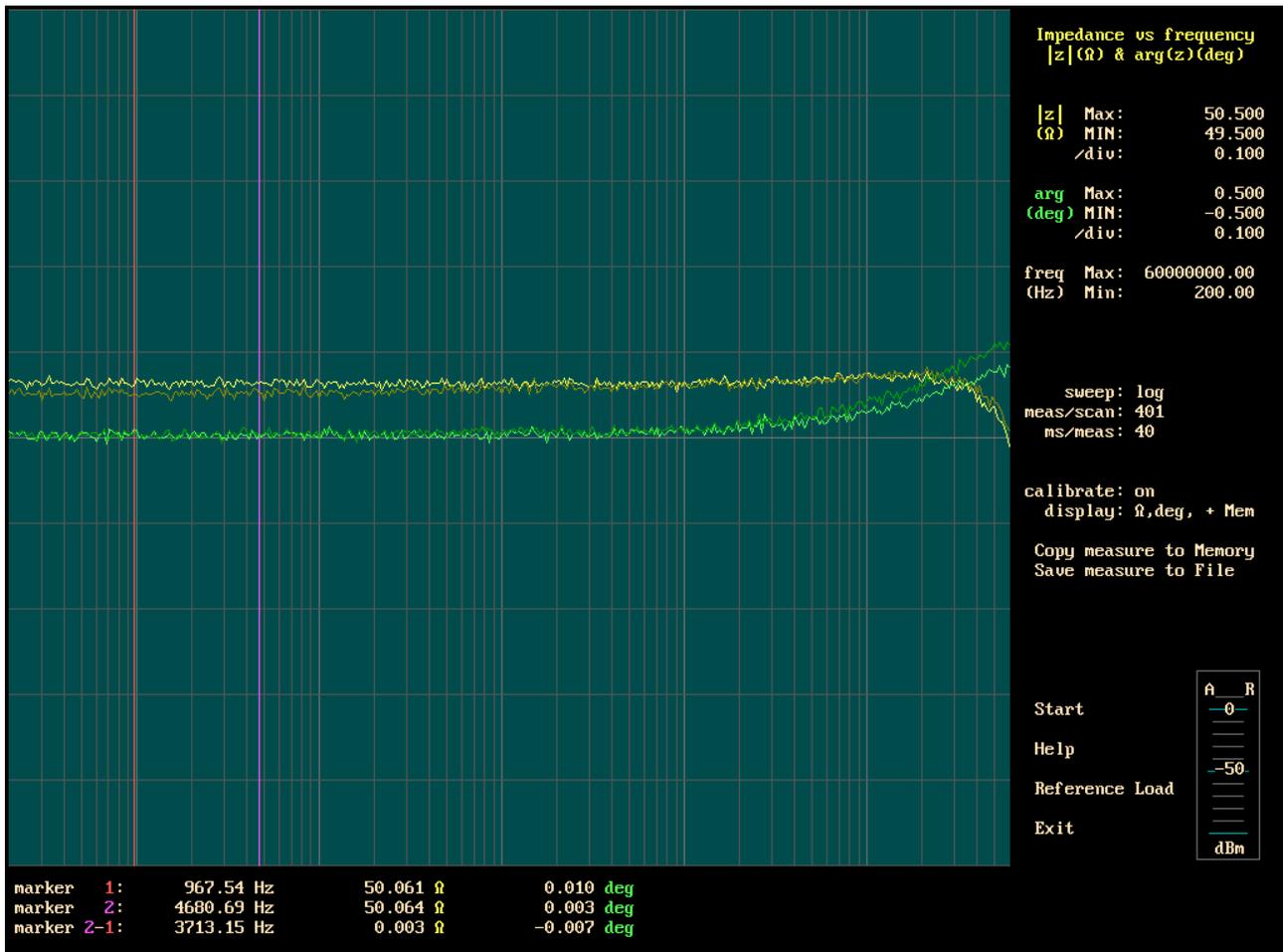


## Impedance measurement on the (A) and (R) inputs (in the s11 format)

To measure these impedances, we need two network analyzers ; (we built only one (150MHz) network analyzer) , so we can only measure up to 60MHz.

At low frequencies, s11 is defined by the precision of  $R1+R2=50$  ohms.

Above 10 MHz, a small capacitor (about 2.7pF in parallel with R1) can improve the value of s11.



## Impedance measurement on the (A) and (R) inputs

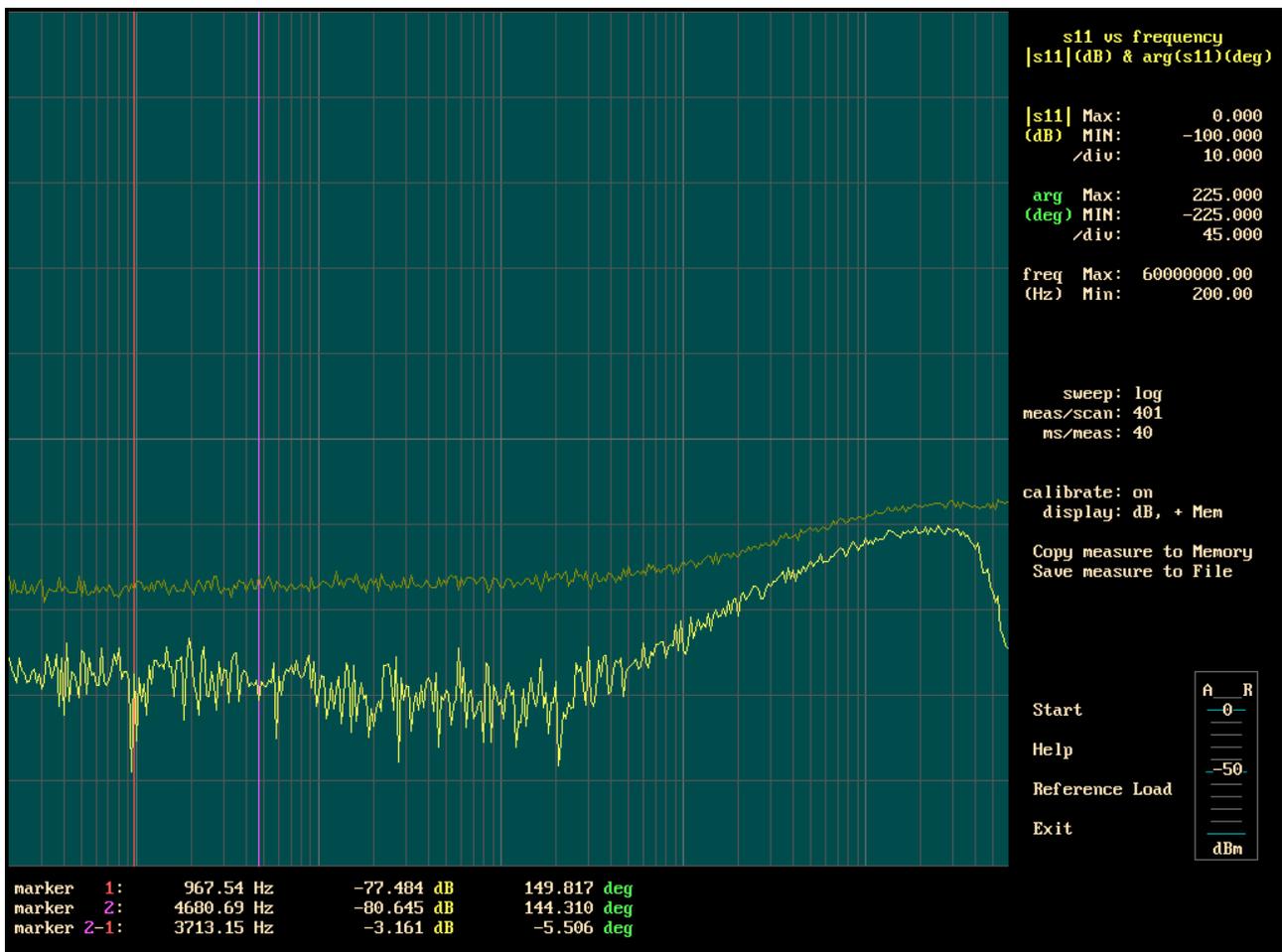
Here is the same measurement as the preceding one; but this time we choose to display the module and the argument of the impedance.

The scales are :  $0.1\Omega$  per division and  $0.1^\circ$  per division.

These measurements are valid only if the reference resistor (used for the calibration) is well defined.

The uncertainties on our measurements are higher when the frequency is high.

We do not have access to commercial analyzers to compare our measurements.



### Impedance measurement of the A and R outputs (in the s11 format)

To measure these impedances, we need two network analyzers ; (we built only one (150MHz) network analyzer) , so we can only measure up to 60MHz.

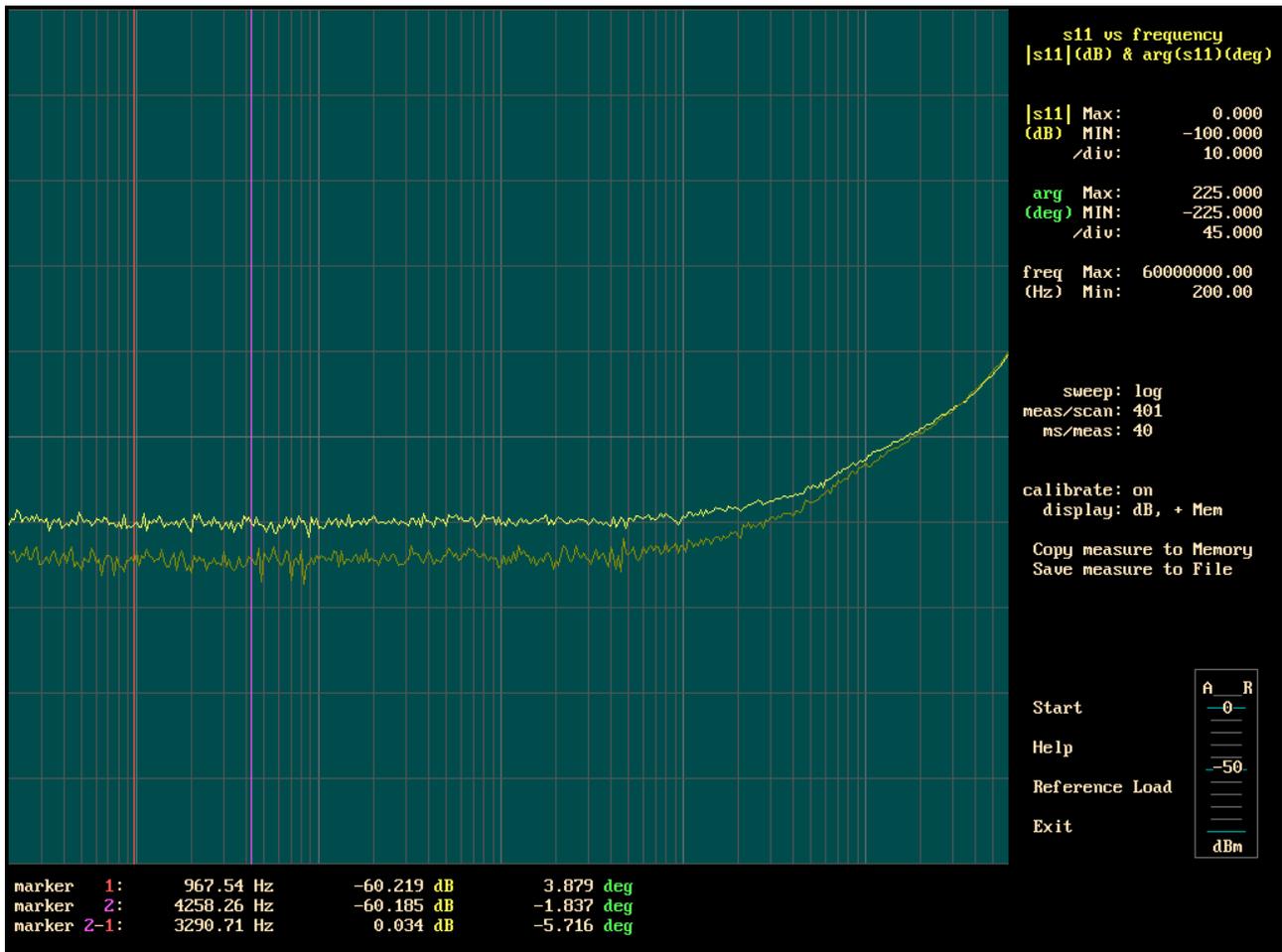
For the network analyzer application, the output impedance of the OPA695 amplifier does not add error on the measurement of the gain vs frequency.

Then, to do this measurement : remove the power supply of the analyzer to be measured, connect to the ground the output of the amplifier OPA695, then measure , (do not forget to remove the short circuit after the measurement !)

The capacitors in parallel with the 50 ohms output resistors are 0.75pf

(the value of the capacitors can change with the way used to connect the coax cables to the board.) .

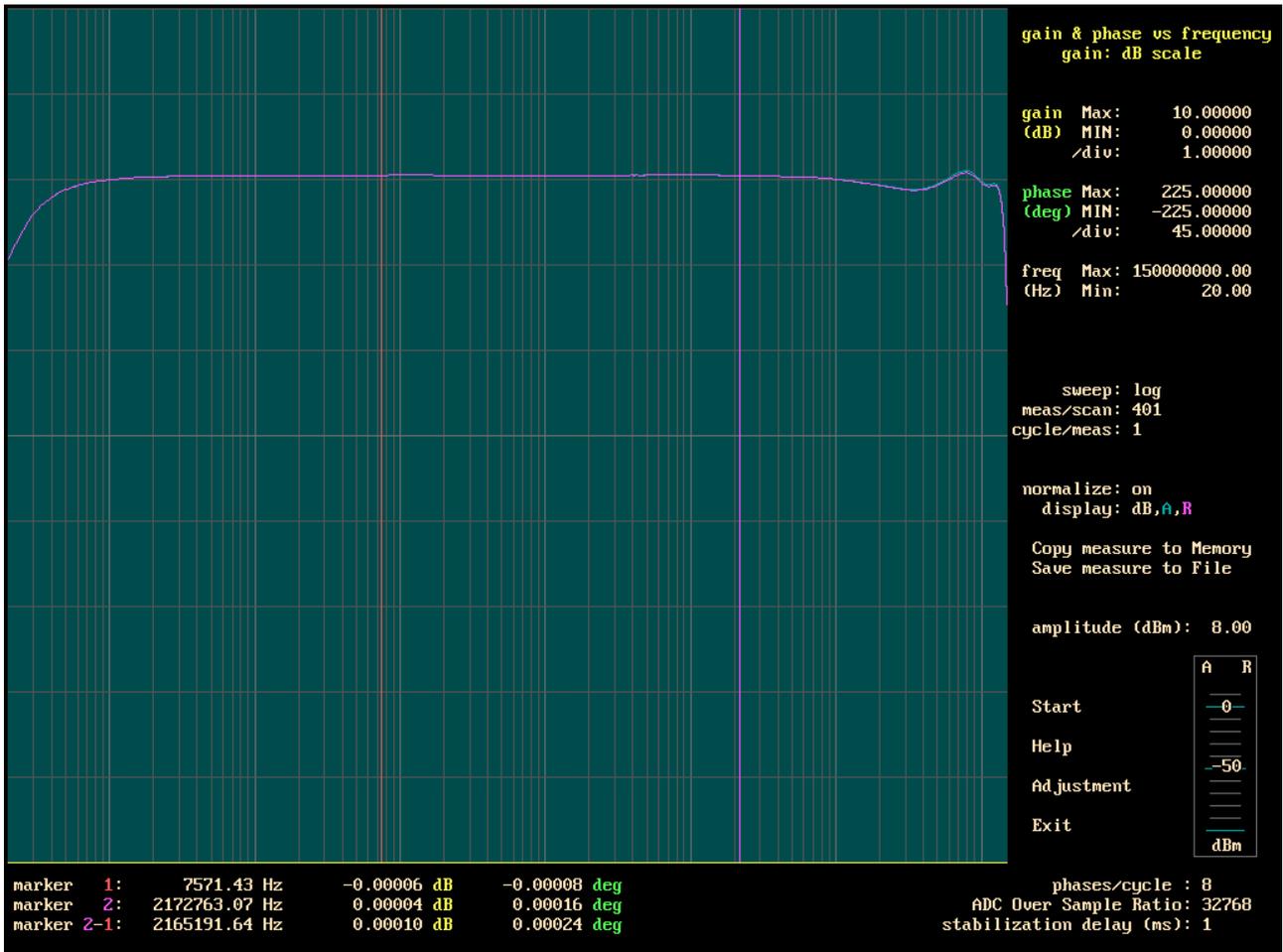
These capacitors are optional , it is just an improvement.



### Impedance measurement of the (A) or (R) outputs (in the s11 format)

If we used the analyzer outputs as a general purpose generator, then output impedance of the OPA695 amplifier contributes to the output impedance.

To make this measurement, connect the power supply of the analyzer to be measured and set the output power of this analyzer at -40dBm, and with another analyzer measure the output impedance.



### (A) and (R) generators output signal level.

On this picture, the purple and blue traces show the variation of the voltage at the output (R) and (A) (measured in dBm).

The scale is identical for dB and dBm. The scale is 1dB/div.

The frequency range is from 20Hz to 150MHz.

At 20Hz, the 1 dB decrease comes from the 330uF capacitors (present at the output of the 150MHz low pass filter and on the emitter of the BRF520 transistor).

When the frequency is high, the gain of the modulators is not completely flat, there is some uncertainty on the measurement displayed.

A parallel resonant circuit (22nH, 47pF, 51Ω) helps to keep constant the amplitude up to 150MHz.

The network analyzer has a good precision to display the ratio (A)/(R), but it has a bad precision to display the signal level on the (A) or (R) input.

The markers measure only the ratio (A)/(R).

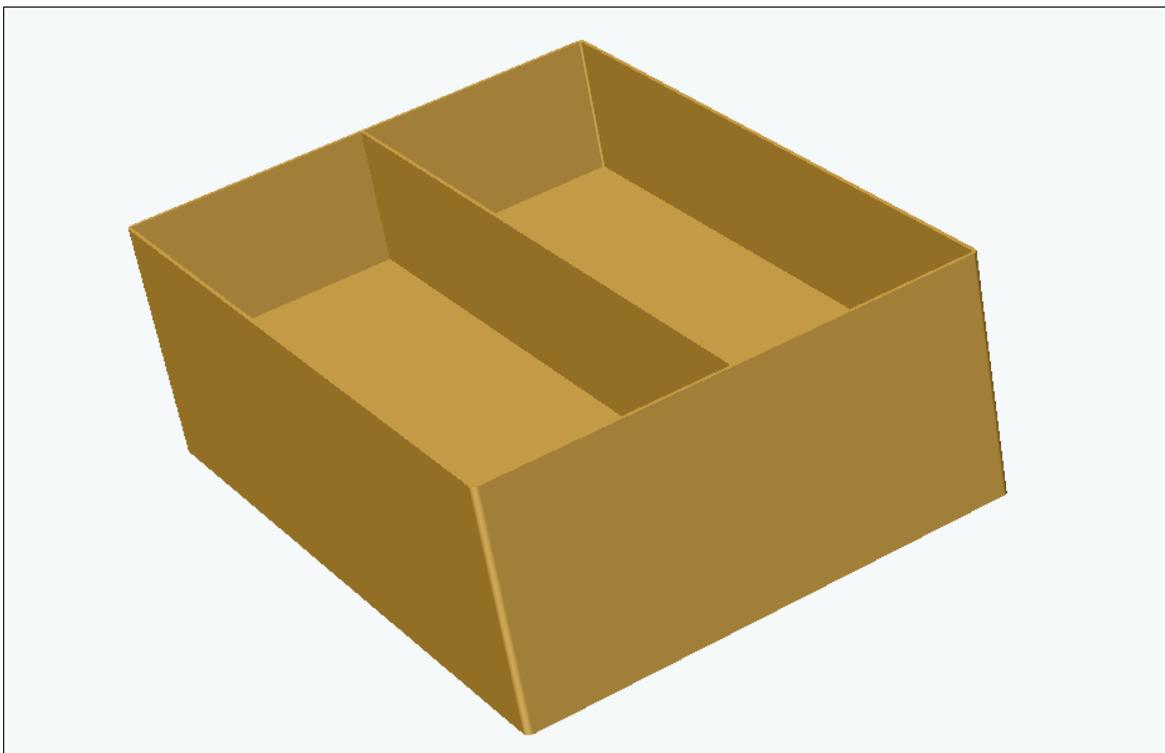
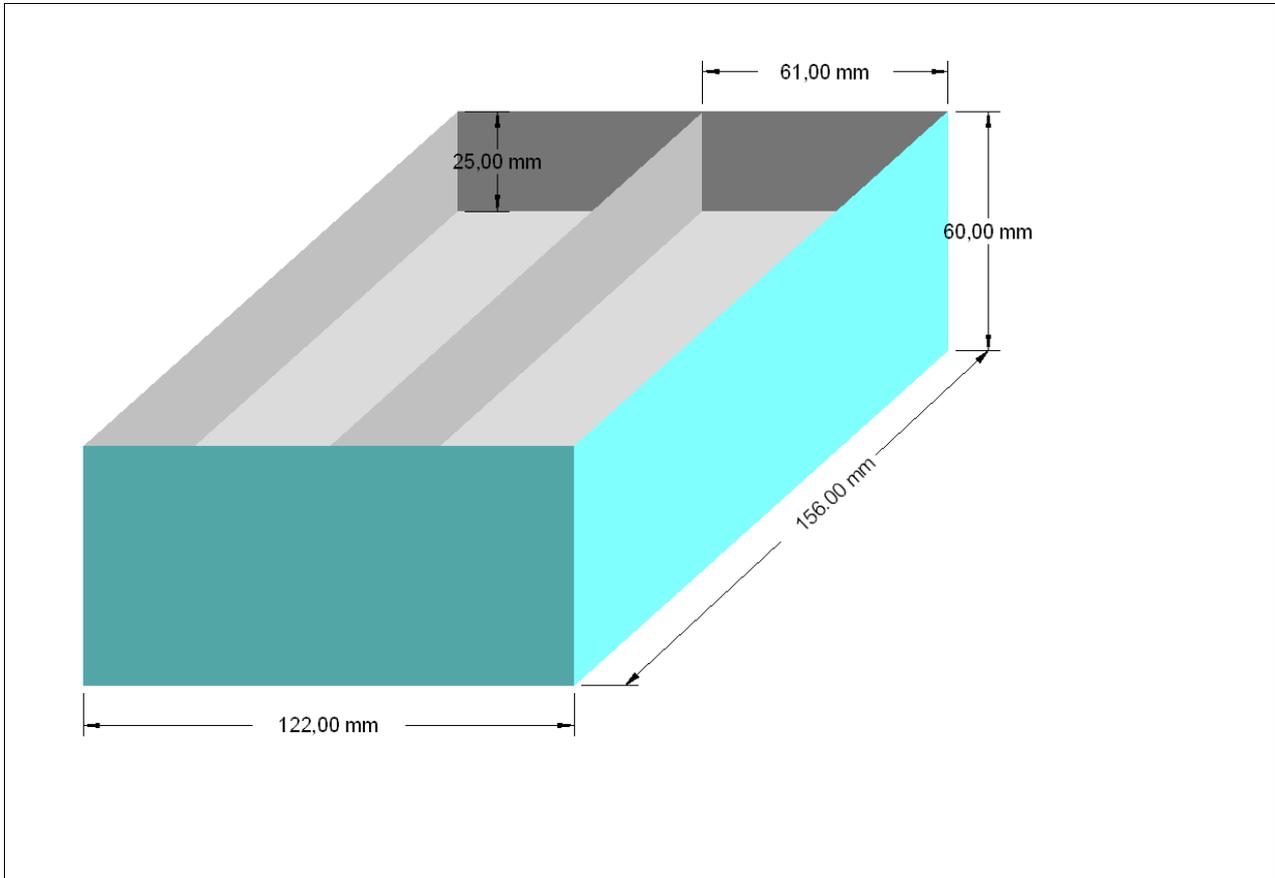
### To measure at a frequency lower than 1000 Hz

It is possible to make measurements from 20Hz. However, to measure the low frequencies; it is necessary to increase the value of the OSR. When the OSR increases the scanning rate decreases and the measurement can take a lot of time.

## The box

To have more than 100dB of dynamic range, it is necessary to separate the detector (A) and the detector (R).

The generator DDS1 and DDS2 are also separated.



The outer wall of the box is made with an iron sheet, dimensions 551mm\*60mm\*0.8mm

There is a bend in each corner.

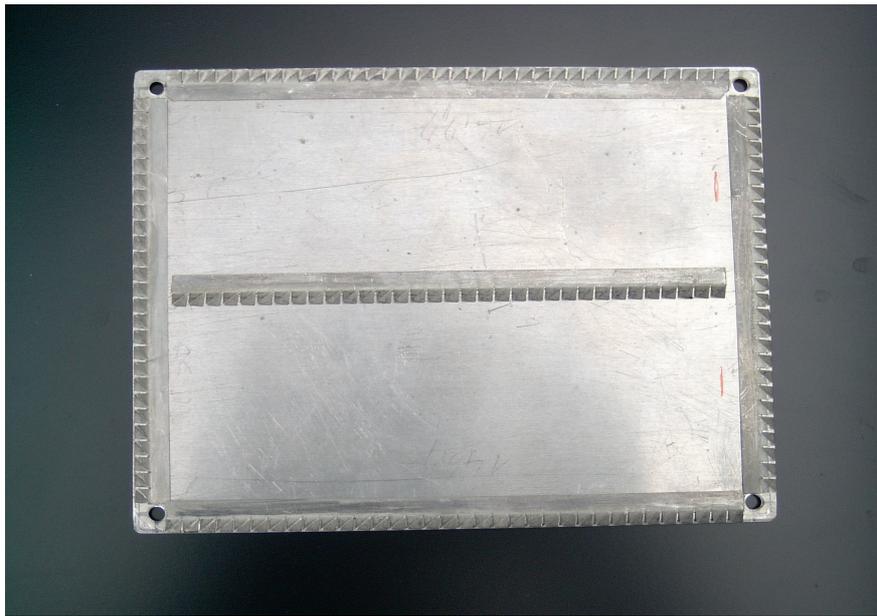
The exact length of the sheet depend of the bending radius. The junction is in the middle of the back side.

The vertical dividing wall, and the two horizontal dividing walls, are made with 3 iron sheets, dimensions : 154mm\*60mm\*0.8mm.

The top and bottom lids are made with 2 aluminium sheets, dimensions : 124mm\*158mm\*2mm

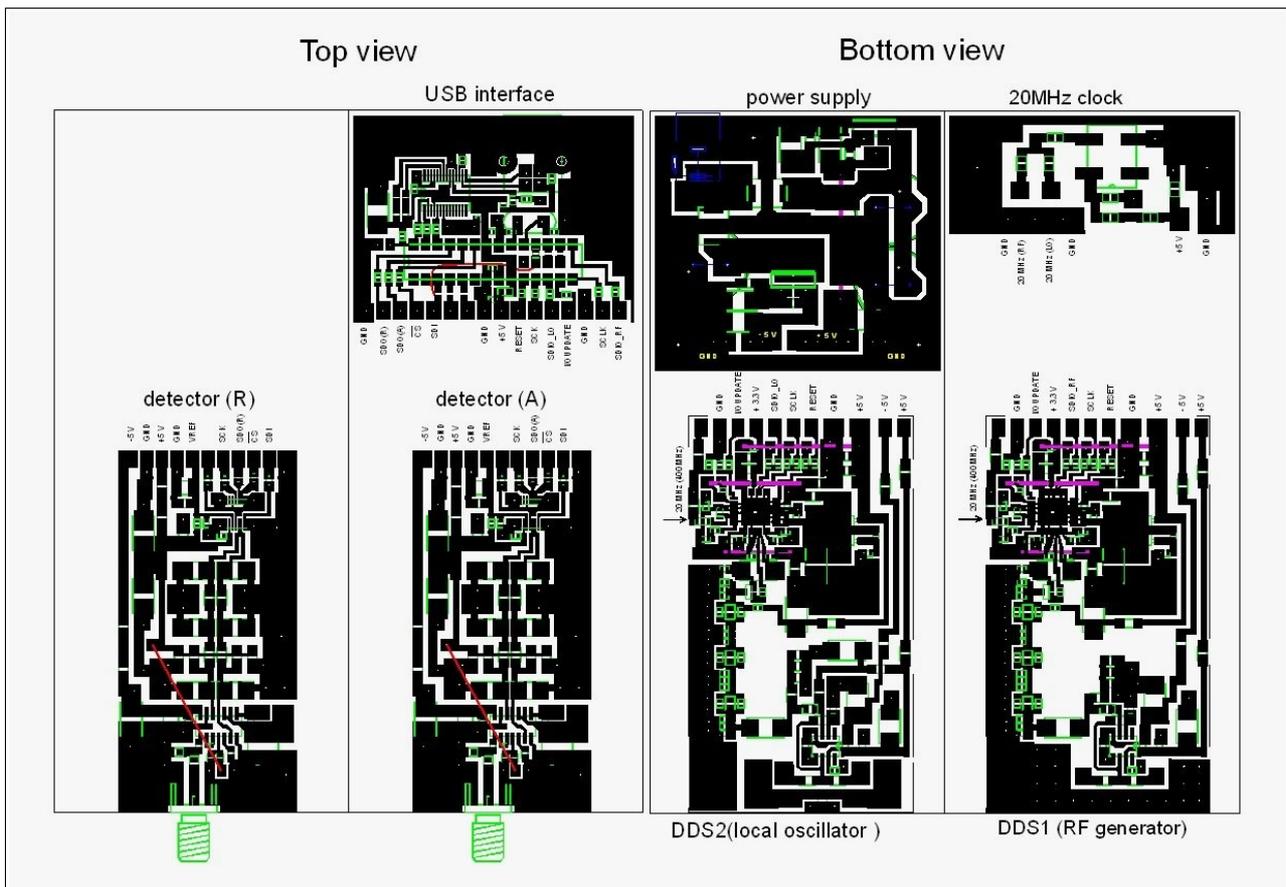
On the lid, a deformable metal joint ensures a certain insulation between the channel (A) and (R).

The thickness of the lid (2mm) is not sufficient; the lid becomes deformed a little. A thickness of 3mm would be better.



Iron sheets are soldered with a soldering wire used to solder electronic components.

We used (iron + nickel) sheets, because we had this sheets; but tinned iron is less expensive.



**Disposal of the printed circuits inside the box**

**Attention !!!**

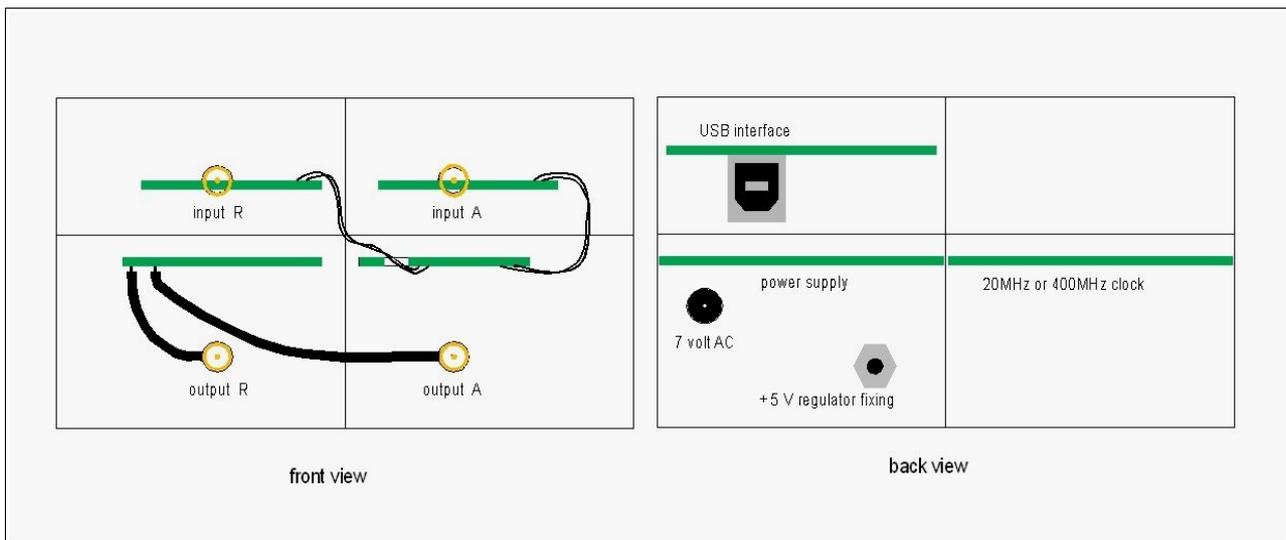
The analyzer currently carried out corresponds to the electrical schematics presented. On the version of the analyzer that we realized, certain components of DDS1 and DDS2, are placed on the ground plane side (transistors BFR520 and several 330uF); (a lot of modifications have been necessary to realize the analyzer !).

The drawing above corresponds to a new layout of the components on the printed circuits (not yet realized).

On the new layout, all the components are on the top (for DDS1 and DDS2).  
On the detector boards, there are only some small modifications

The probability to have some problems (with this new layout) are low because the electrical schematics are identical, and the disposal of the critical components is not modified.

For the moment, we have no other AD9951, we hesitate to unsolder the AD9951 of our prototype to resolder them on the new study (we are afraid of damage during transplantation !).



### Disposal of the printed circuits inside the box

## Atmega168 programming

To program the micro-controller there are two possibilities :

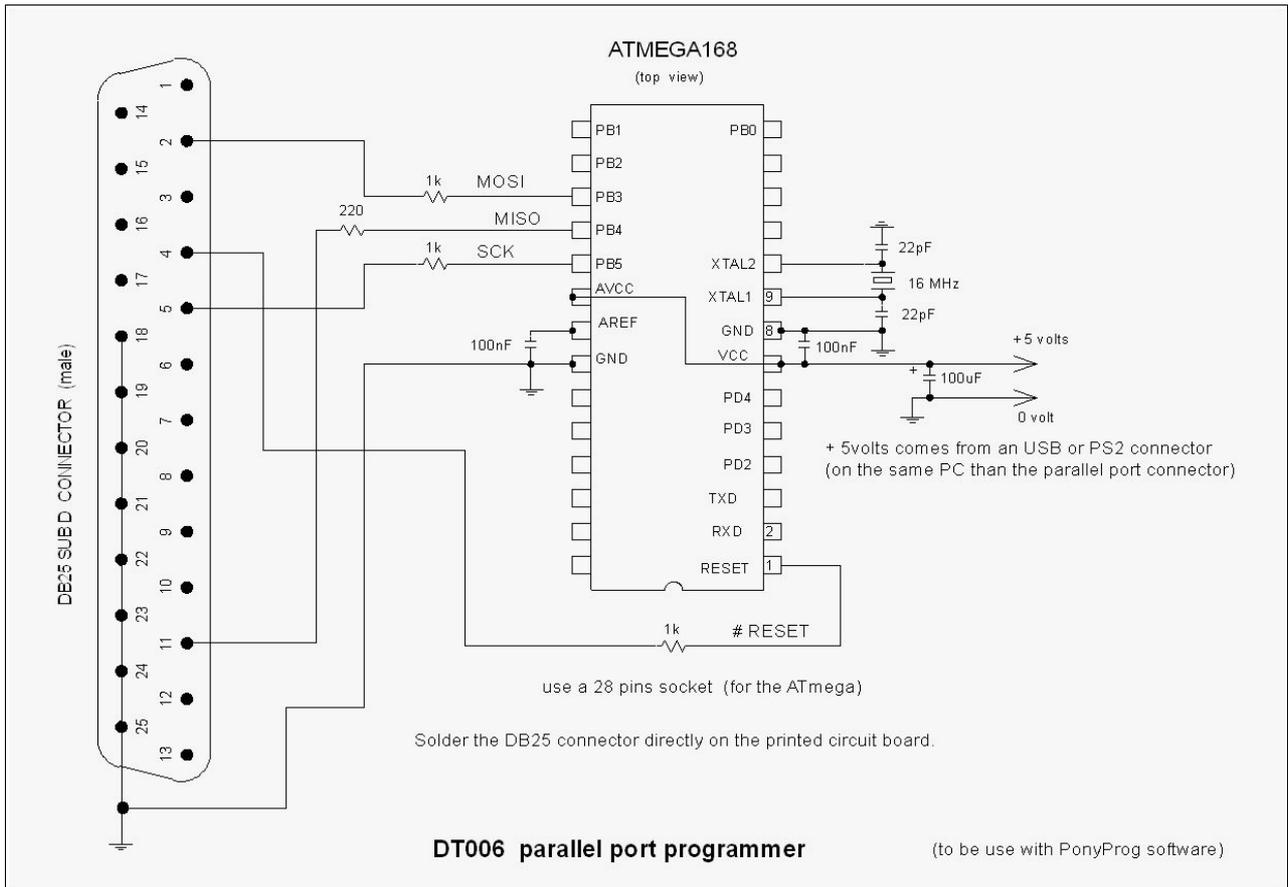
1 - Buy a Diecimila board, and load the network analyzer program with the Arduino software (through the USB port).

2 – Buy an ATMEGA168 and program the flash memory with a parallel port programmer. The Arduino project has a parallel port programmer, but we do not succeed with this programmer ? We have used a parallel port dongle DT006 with the PonyProg software .

<http://www.lancos.com/prog.html>

The DT006 dongle is very easy to built : 4 resistors and a parallel port connector.

Under Windows, the PonyProg software needs to access the parallel port directly (this is not possible with 2000 and XP); it is necessary to install a software like “userport” or “Porttalk” to allow this direct access.



In the PonyProg menus choose : AVR micro , Atmega168 and DT006 parallel.

If there is nothing in the flash memory, when we read it, there is FF everywhere.

With the PonyProg software, write the file xxxxxx.e2p (in the Atmega168 flash memory).

Then write the **configuration and the security bits** (write the flash memory before the security bit).

**Configuration and Security bits**

7  6  BootLock12  BootLock11  BootLock02  BootLock01  Lock2  Lock1

7  6  5  4  3  BOOTSZ1  BOOTSZ0  BOOTRST

RSTDISBL  DWEN  SPIEN  WDTON  EESAVE  BODLEVEL2  BODLEVEL1  BODLEVEL0

CKDIV8  CKOUT  SUT1  SUT0  CKSEL3  CKSEL2  CKSEL1  CKSEL0

Checked items means programmed (bit = 0)  UnChecked items means unprogrammed (bit = 1)

Refer to device datasheet, please

Configuration and security bits