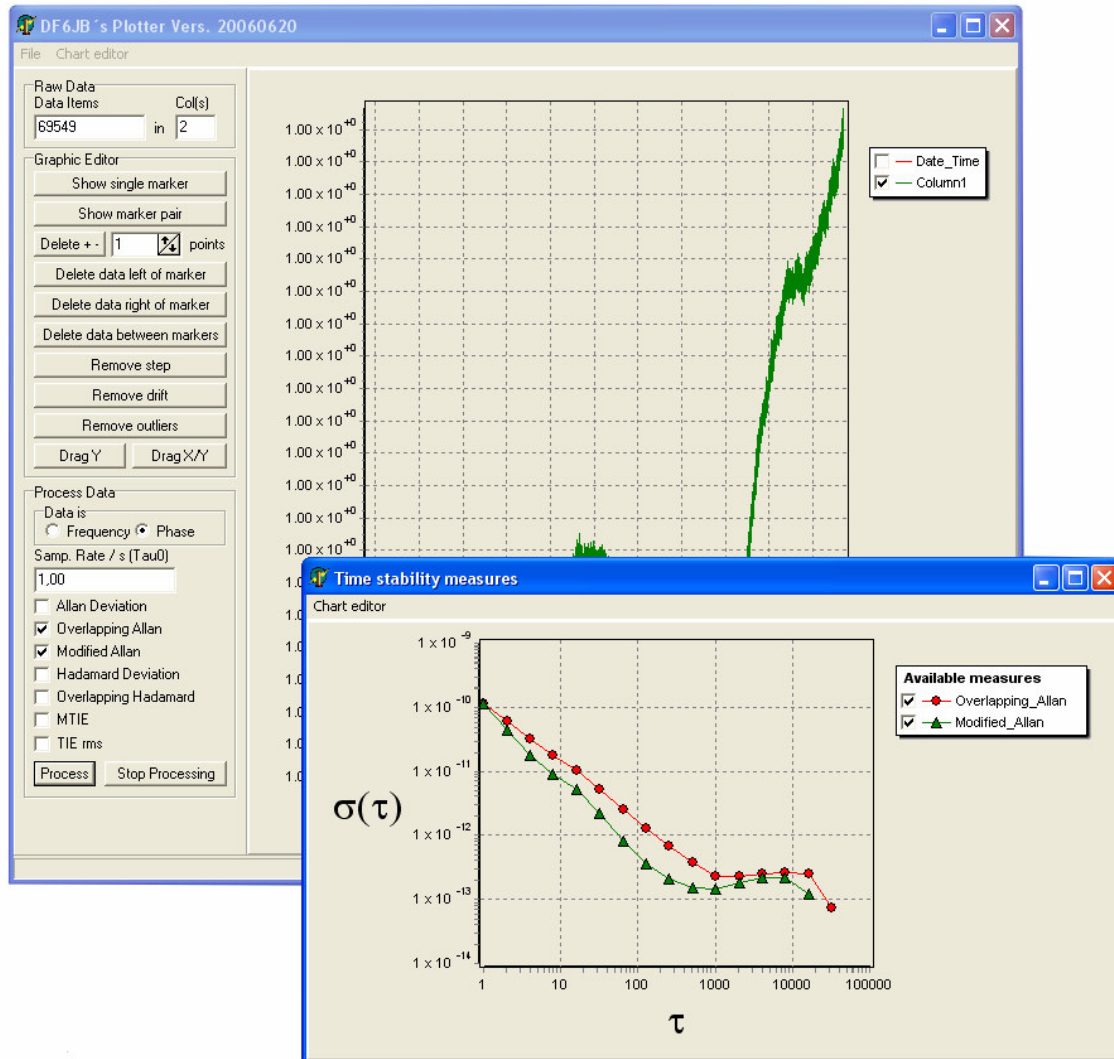


DF6JB's Plotter



A program for the graphic manipulation of data and for the computation of some statistical properties used in time stability analysis

Ulrich Bangert
df6jb@ulrich-bangert.de
Ortholzer Weg 1
D-27243 Gross Ippener
Germany

Introduction

Thanks to Bill Riley of Hamilton Research I am in the lucky position to have a copy of STABLE32 running on my computer to make my oscillator stability computations with. STABLE32 is the de facto standard in horology and its only drawback is that not all of you time nuts have a copy of it available due to the fact that excellent products do have its price. On my search for an alternative to STABLE32 I came over ALAVAR5.2. This is a freeware tool that you will easily find on the web, just google for it. ALAVAR does the statistical math nicely but lacks completely any data editing capabilities. Data editing capabilities? Why should your statistical software have data editing capabilities?

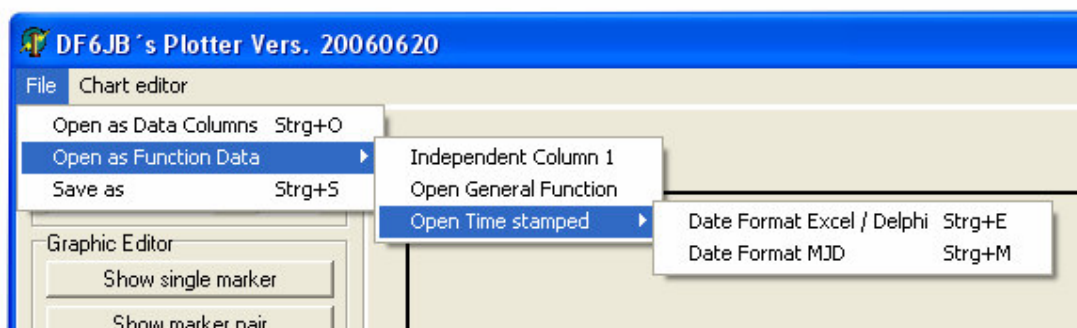
Well, over the last years I have found that computing statistical properties of data is one part of the job and that finding and eliminating data glitches and outliers is a second part of the job, sometimes *a lot more time consuming* than the math itself.

Given your statistics software can show you outliers in your data at all: In most cases you will have to leave the statistics software, enter a text editor, remove the outliers and start from the beginning. In other cases, for example your phase data may suddenly show an offset within a measurement, even the text editor won't be a help because now you need to manipulate *lots of data points*.

That made me think that easy graphical manipulation of data is as important as the statistical math itself and because I could not find anything that did exactly what I thought how it should be done I started to write my own software. What you keep in your hands now is the first try to let someone else use this software than me myself. It is far from being perfect and any feedback from you concerning errors or possible improvements is highly appreciated!

Using DF6JB's Plotter

Once you have started **Plotter** you end up with an empty graphics display because you have not loaded any data yet. Use the **File** menu entries to load data.



Data is always expected to be read from an **ASCII text file**. That is why the default extension in the standard file open dialog is “.txt”. Other extensions are allowed but keep in mind that basically it is text what **Plotter** expects.

After you have chosen a specific file, **Plotter** will try to read it. **Plotter** knows a number of different delimiters that are commonly used in text files. It is also aware of the fact that different countries use different **decimal delimiters** and it will try it's best to handle everything correct but sorry: there are situations where **Plotter** cannot automatically decide what to do. If you ever run into the situation that a text file cannot be read with **Plotter** correctly, send me a few lines of the file via email and I will try to check for it and perhaps improve **Plotter**.

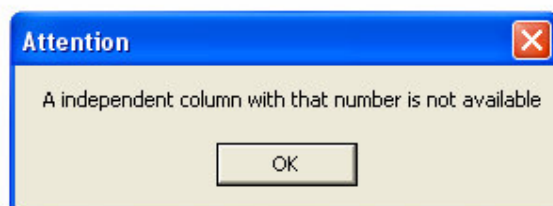
Data may be read in three different modes:

- 1) **Open as Data columns**
- 2) **Open General Function**
- 3) **Open Time stamped**

In the first case every column that is found in the text file is expected to be a simple data column. The horizontal axis the data point's number displays in this case.

In the second case one of the data columns is expected to hold the independent variable of a functional correlation and all other data columns are expected to hold data that is dependent on the independent variable. In this case the independent column is loaded as a data column like the others too, but the data of the independent variable is also used to generate the horizontal axis data from which gives a true X/Y display.

Note that the number of the independent column may be chosen free by the menu entry **Independent column**. If you set this to a wrong number, loading data will result in the error message



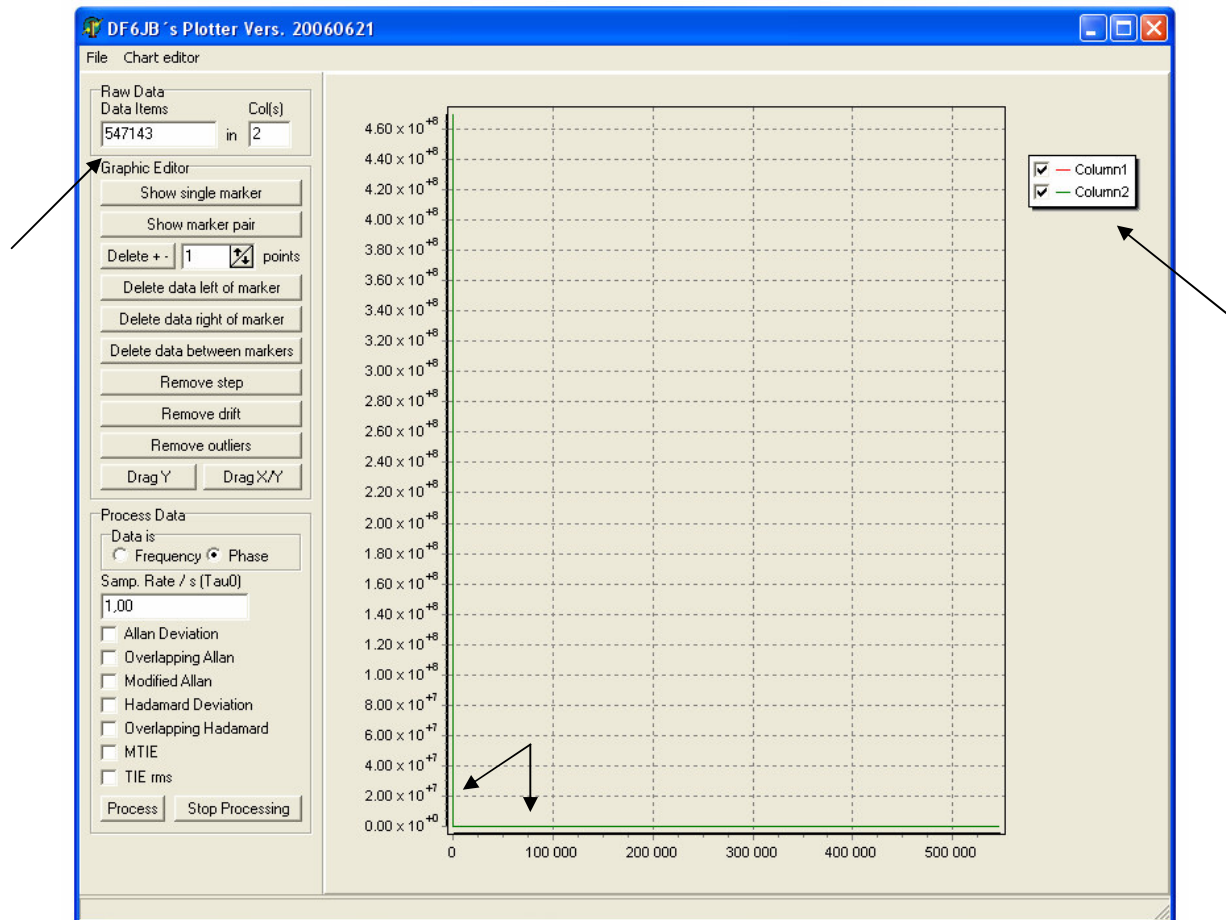
Open Time stamped is a special case of *Open General Function*. The independent variable here is **time**. This case is handled on its own because 2 different formats are commonly used for date/time information. Also for date/time the number formatting for the horizontal axis is different from “normal” numbers.

Time stamps may be in the “Excel / Delphi” format or in the MJD format. In the “Excel / Delphi” format a time stamp is a floating point number which's integer part is the number of whole days since January 1 of 1900 and the fractional part contains the fraction of the day, i.e. the time of the day. High noon = 0.5.

MJD stands for “Modified Julian Day” and simply uses a different “day one” reference which is more common among scientists. If your data looks something like this:

5.38783570790854E+0004 9.99999488619718E-0001
 5.38783570919214E+0004 9.99999488535211E-0001

it is most probably Ok! In case you have read time stamped data the graphics display's horizontal axis will show you date and time of the data. Otherwise it will show you the number of data points. Of course you may read also time stamped data simply with **File Open as Data columns**, you just have to remember yourself that the first column in this case contains no data but date and time.

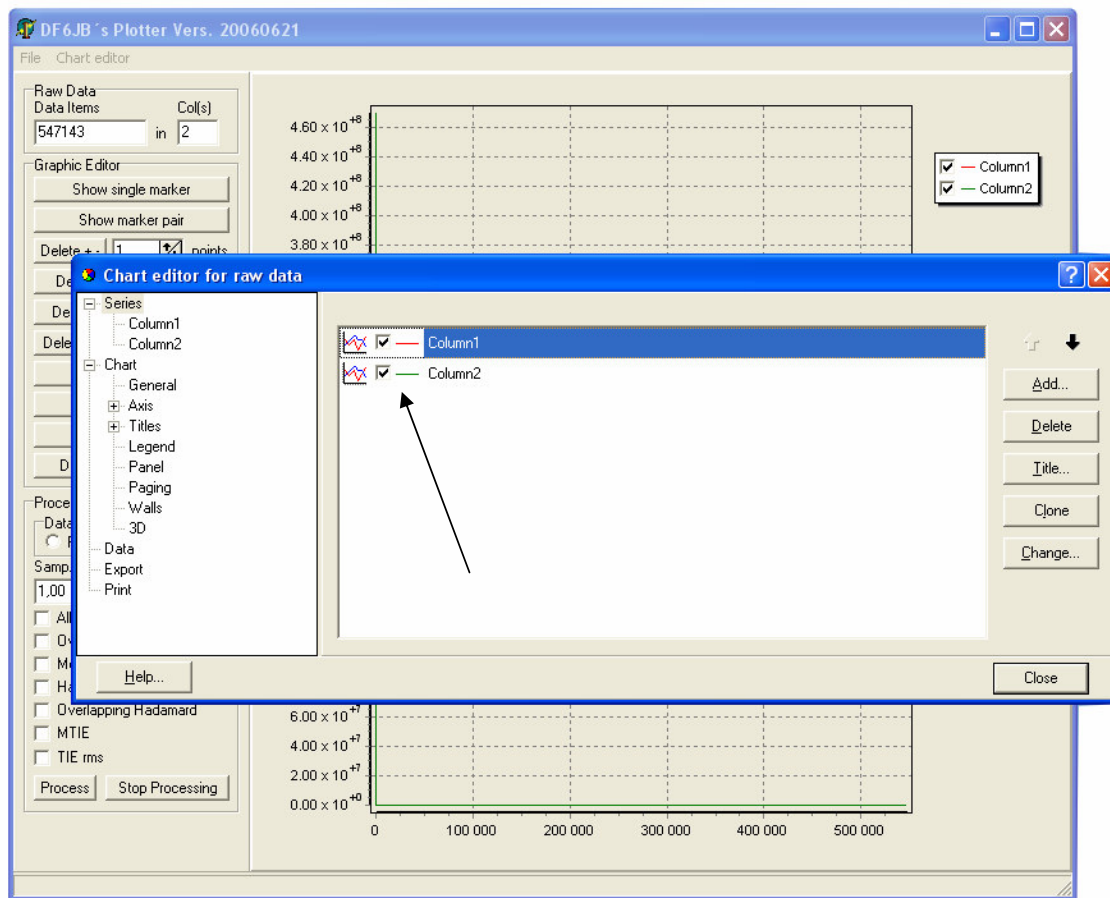


Once you have loaded data correctly, **Plotter** will inform you about the number of data items it has found in the file and the number of data columns (left arrow). It also has given an individual name to each data column (right arrow) and tries to display all of them in the graphics display.

Since all data columns use the same vertical axis (the left one) and may contain outliers at this time, *do not expect the graphics display to look nice immediately after loading data*. We will need a few seconds more for that!

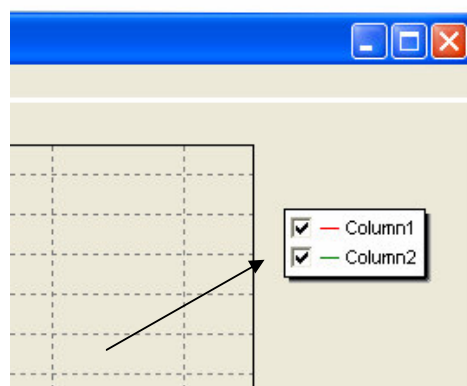
In the example above, data seems to be loaded correctly but is not displayed nicely either due to both columns being displayed or due to one column having a big outlier right at the start (centre arrows). Let us start with looking at only **one** data column.

Open the **Charteditor** from the main menu to see something like this:

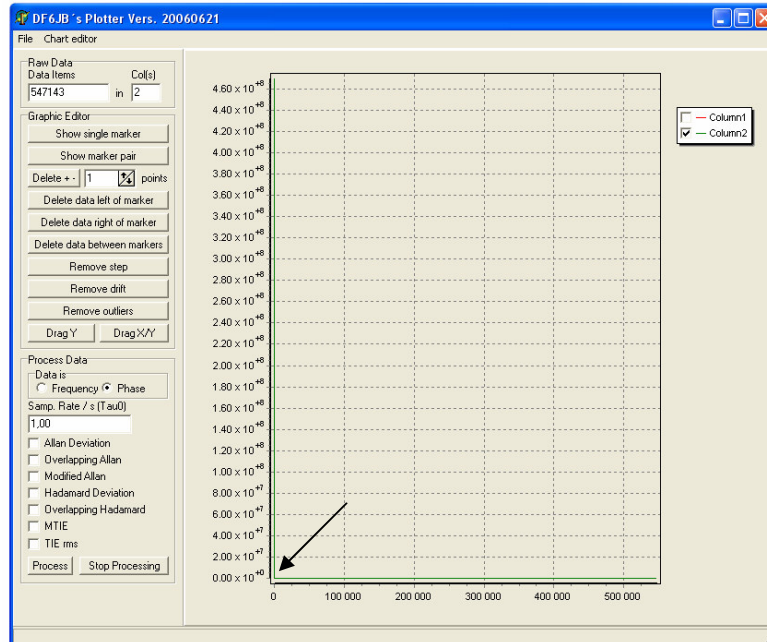


Note that the checkboxes for both columns are checked (arrow), meaning that both columns shall be displayed. Uncheck the first checkbox to make only data column 2 being displayed and close the Charteditor window.

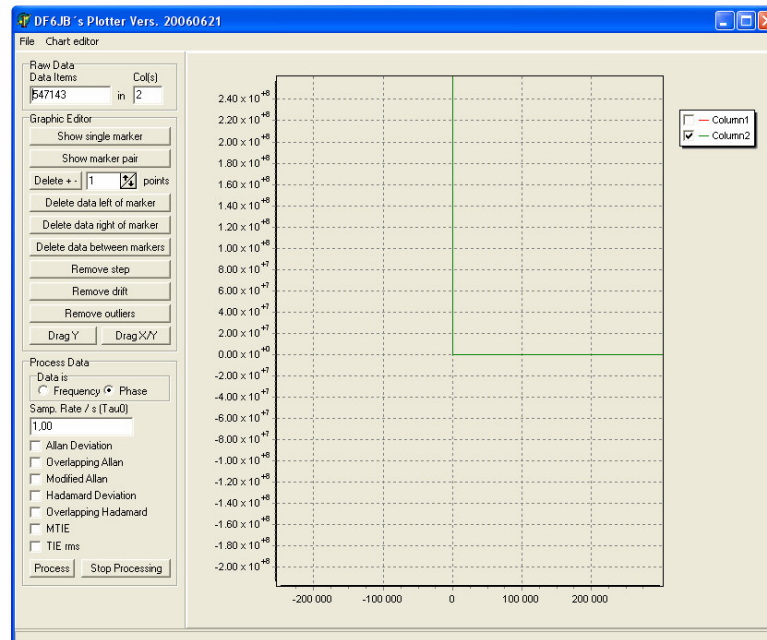
Note that in the current version of plotter the same can be done by unchecking the checkboxes in the main chart's legend, but we will need the chart editor for more complex manipulations.



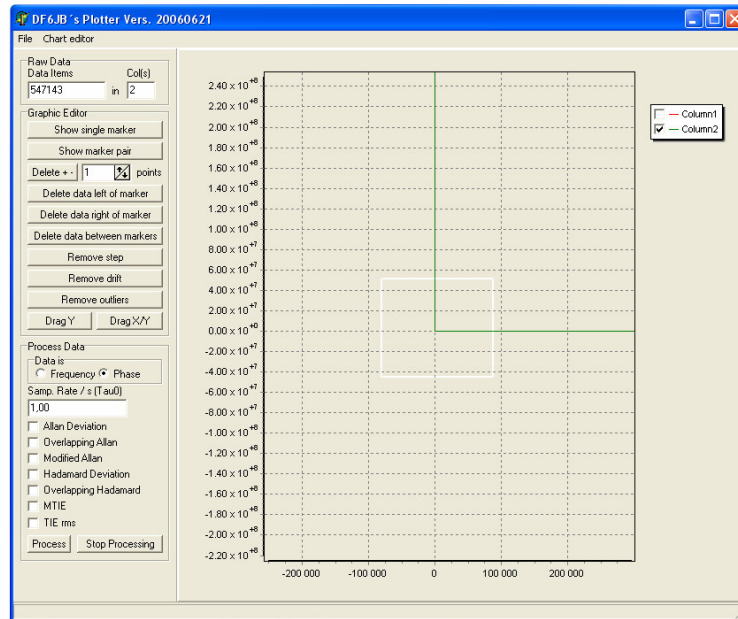
After having done so we end up with the display below. The graphics still does not look nice, but notice the legend has changed: Since only one column's data is display the legend now tells us something about this column's data points and there is one **big** point at the beginning of the column while the other data points are more at 1! Most probably this is an outlier. Let's remove it!



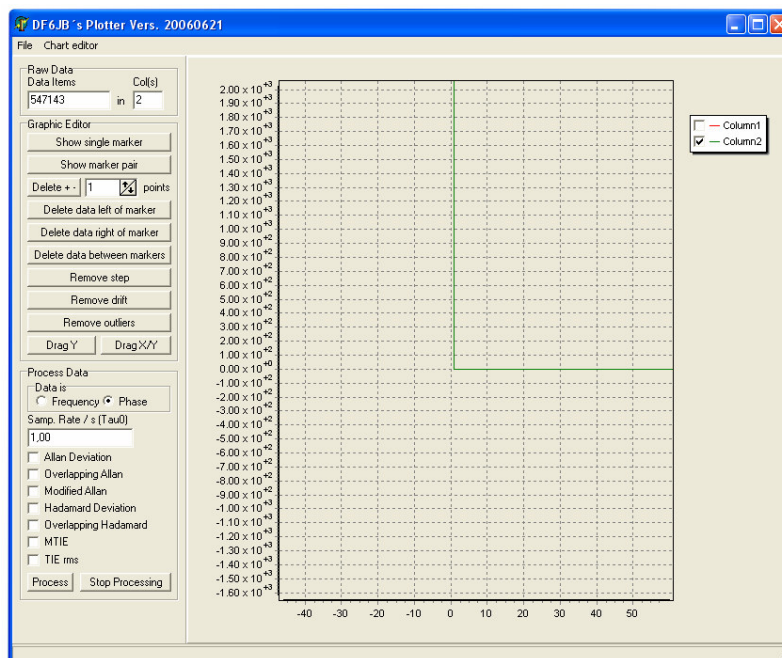
Use the **right mouse button** to click near the left down corner of the graphics display say where the arrow points and **keep the mouse button pressed**. In this mode you can drag the contents of the graphics display. **With the right mouse button being pressed move the mouse to the centre of the graphics display**. The display now looks like



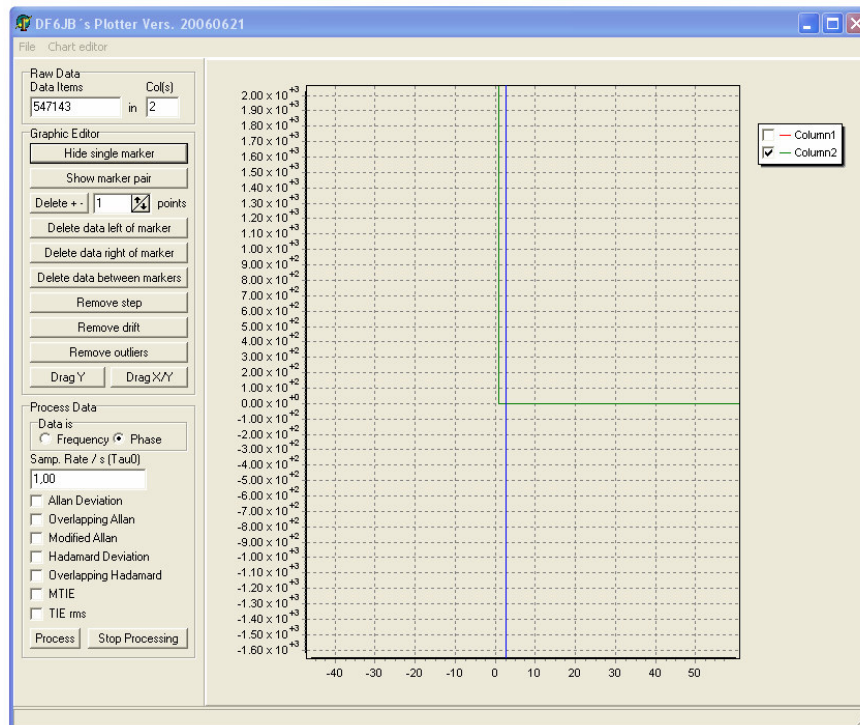
Since we want to zoom now around the outlier it is much more appropriate to have it in the centre of the display. Open a zoom window. This is done *by pressing the left mouse button inside the graphics area and while keeping the left mouse button pressed moving the mouse to a point that is right and down to the starting point*. At the new location you can leave the left mouse button and the area inside the white square will be displayed zoomed



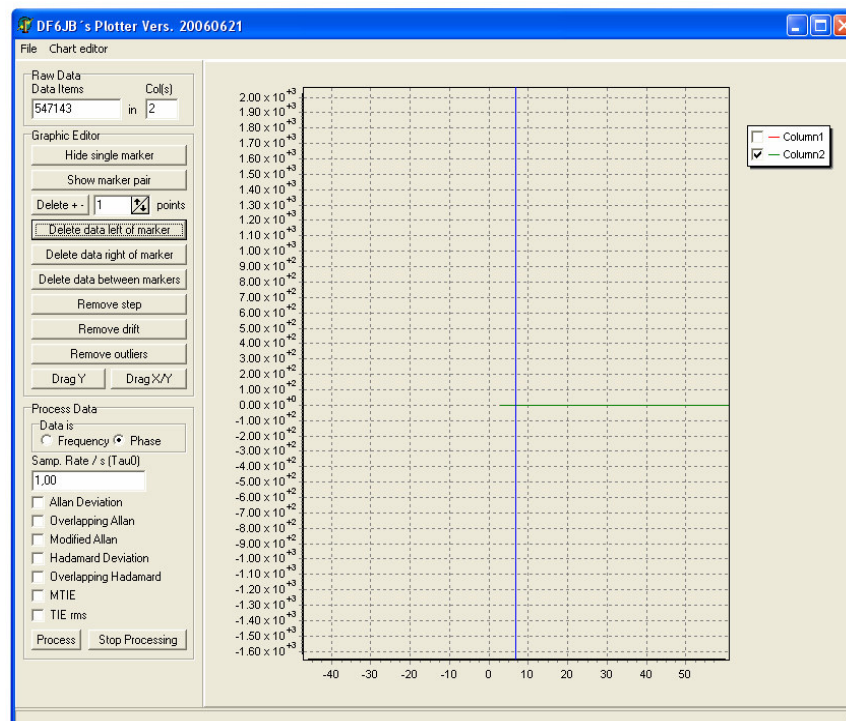
You can zoom in an unlimited number of times. Repeat zooming in a few times until the x-axis tells you that only a small number of samples are being displayed, say less than 50-100 as shown below.



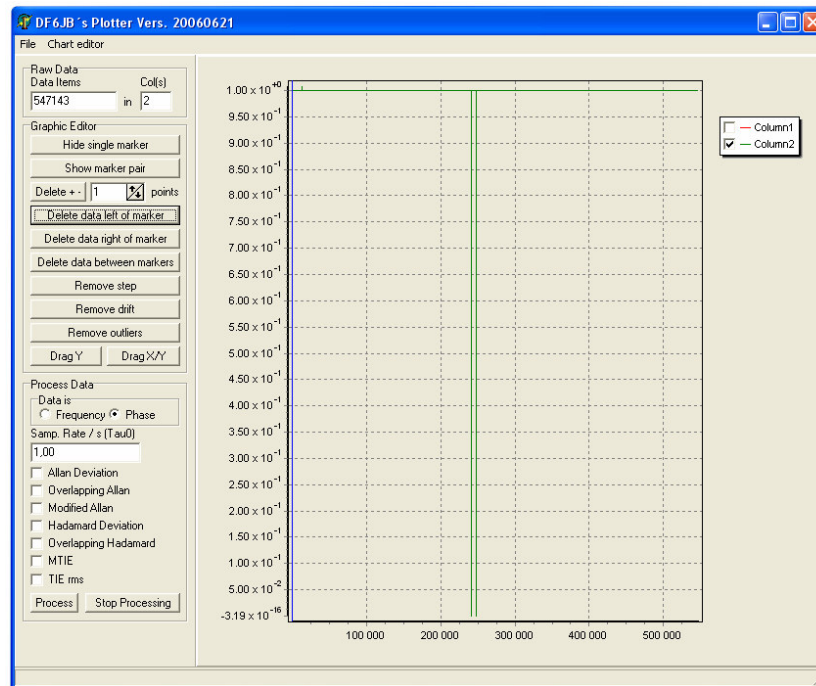
Press **Show single marker** from the **Graphic editor box** left in the window and watch a blue marker line appear in the centre of the graphics display. Grab the marker with the mouse and shift it immediately right from the outlier.



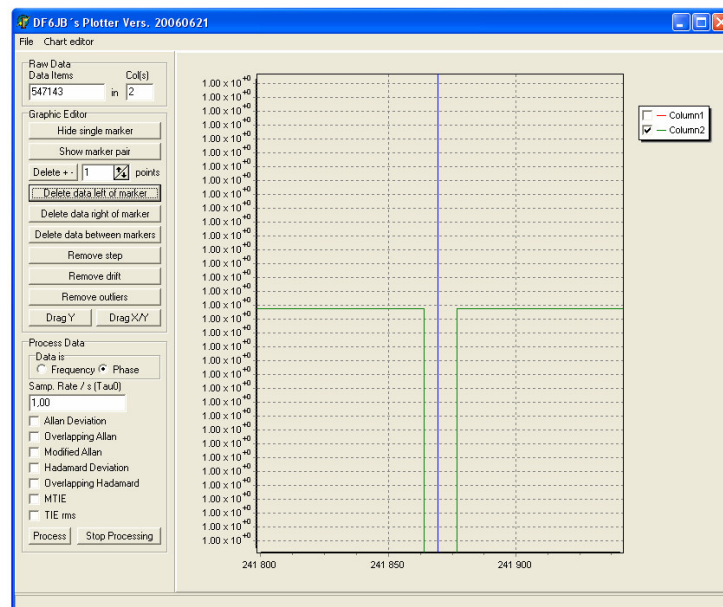
Press **Delete data left of marker** from the **Graphic editor box** and watch the outlier and some other data points disappear from the graphics display.



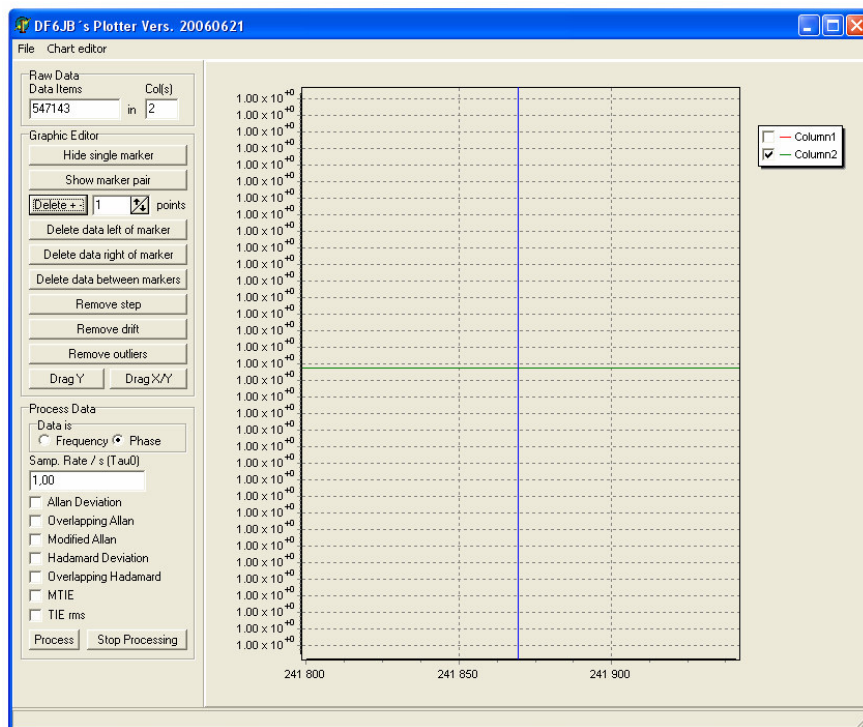
Un-zoom the window and un-drag the window *by pressing the left mouse button and while keeping it pressed moving the mouse to a point left and up of the starting point*. All zooming operations and all dragging operations of the graphics display are cancelled and you see again the whole of the data:



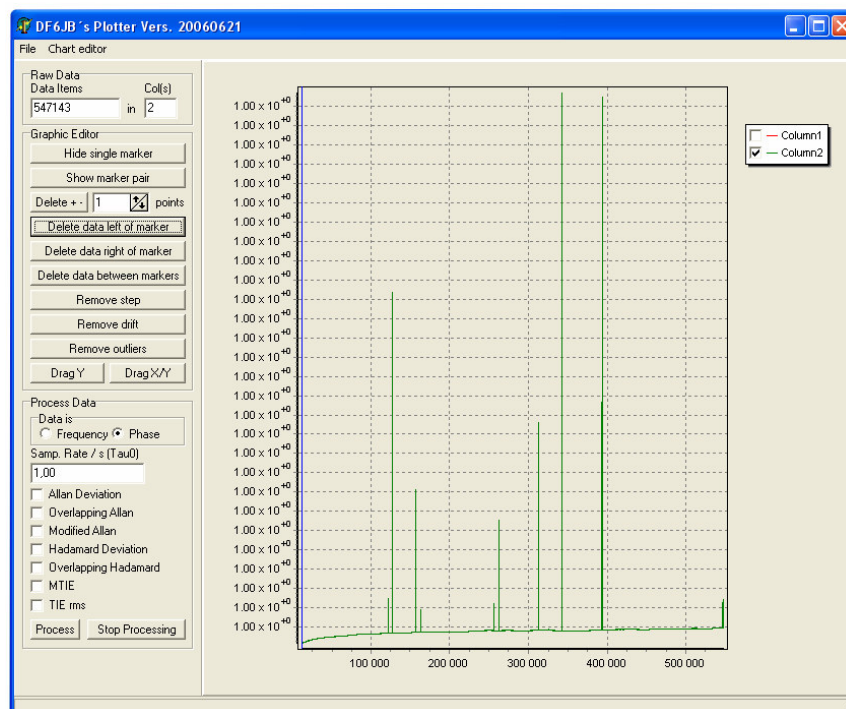
Unfortunately it seems as if there were some other outliers that are too small to be noticed on the vertical scale we used before but big enough to make the data not looking nice on the current scale. Because they are not at one end of the data, we must do something different to get rid of them. Again zoom in a number of times around the point where the data leaves its normal range making the display look like



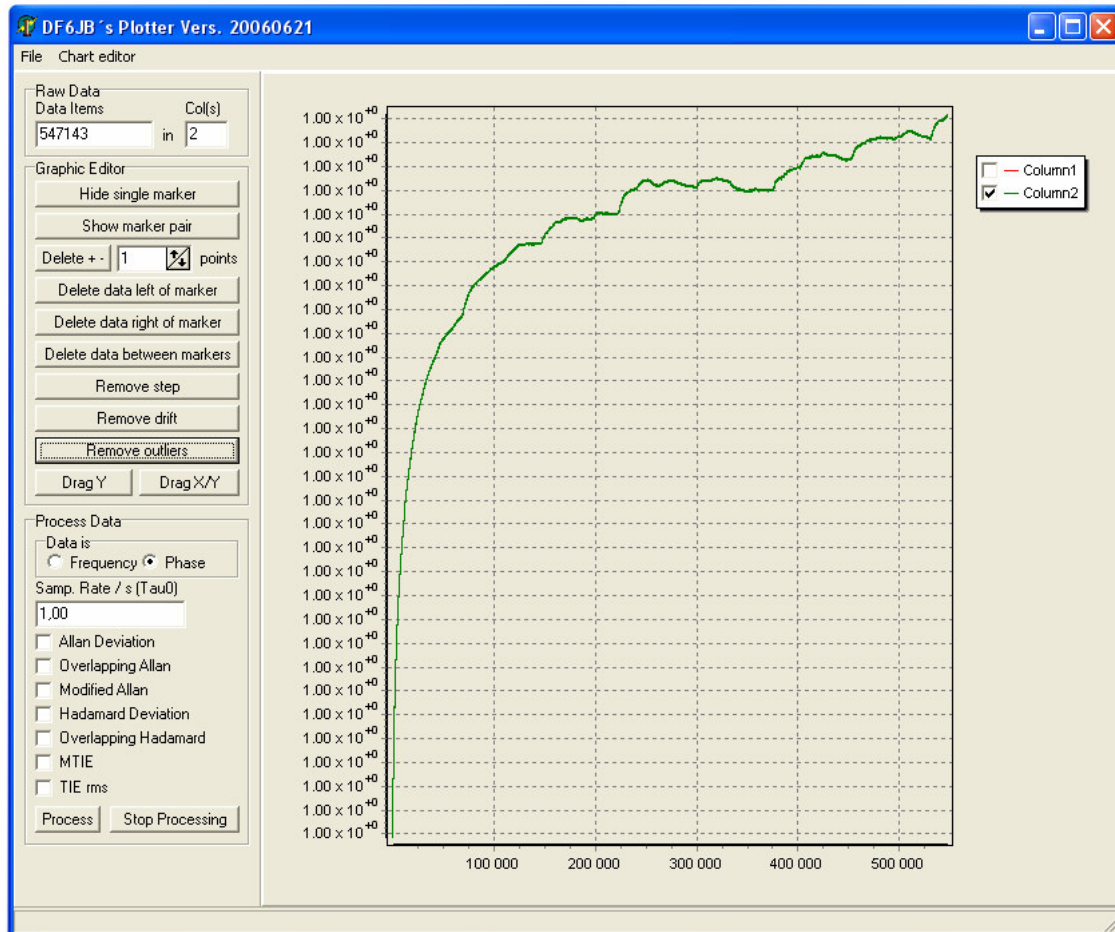
Press **Delete +/- N points** from the **Graphic editor box** one or more times, depending on how many outliers are contained in the peak until the peak has disappeared.



Again un-zoom the window and un-drag the window **by pressing the left mouse button and while keeping it pressed moving the mouse to a point left and up the starting point**. Repeat removing the next outliers using the same method. After some additional work you may end up with something like:

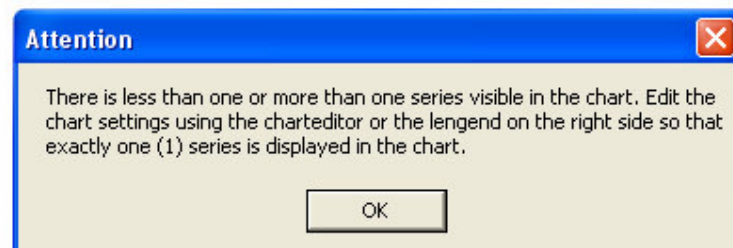


While we see that we are still not through with our work of outlier removing we start to see the real structure of our data column. Note that we can go on with outlier removing as shown before. Note, however, that this has been shown to you just to get you acquainted with the **Graphic editor** features. Instead of removing the outliers “by hand” we could as well just have clicked **Remove Outliers** from the **Graphic editor box** once we switched to the one-column display to receive a result like



That *does* indeed look damned nice! **Plotter** uses some very sophisticated **robust statistical methods** to detect and remove outliers and in general you will be very impressed by the automated outlier detection. There may be, of course, situations where the automatic detection works inferior to what is seen here.

Had we pressed the **Remove outliers** button before switching to the one-column-display a warning display saying

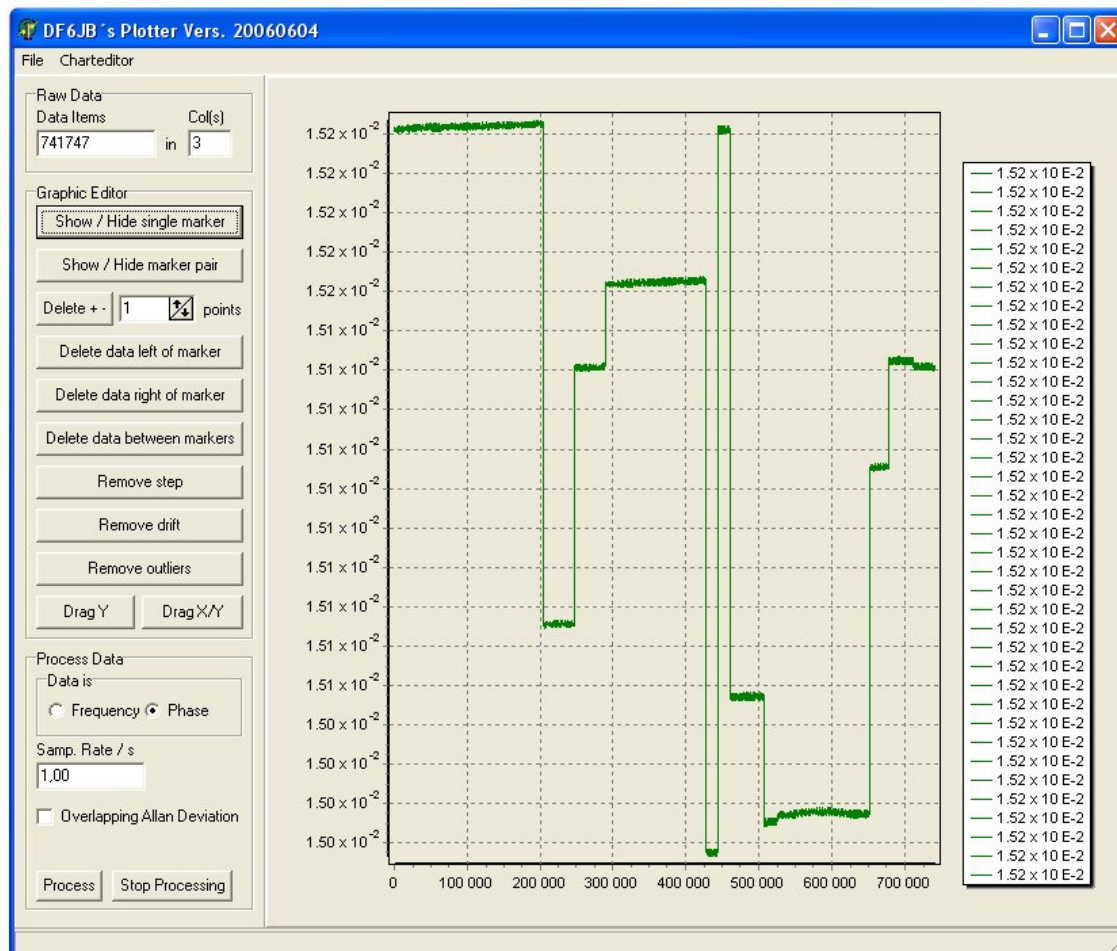


would have reminded us that we request for something that Platter cannot perform.

Before I continue to tell you how to make a professional looking graphic from the already nice looking data, let me explain which other editing capabilities the **Graphic editor** has. In addition to what you did already see you may

- remove all data right of a single cursor
- display a cursor pair and remove all data between the two cursors
- remove a step in data
- remove a drift in data
- drag data points in y-only or x/y direction

After what we've seen before, the first two points are surely trivial, but what does it mean to remove a step in data? Have a look at the following raw data. It shows a 1 pps comparison between my GPS controlled house standard and a DK6RX based frequency standard.

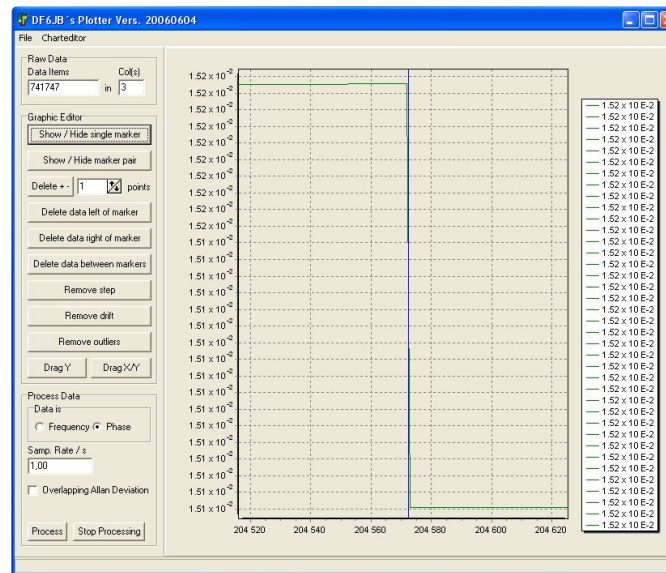


The DK6RX standard is based on the reception of the television channel ZDF (Zweites Deutsches Fernsehen) in Germany. The ZDF's transmission central in Mainz is believed to have a rubidium frequency standard synchronized to PTB (Physikalisch-technische Bundesanstalt, our German equivalent to NIST) which is used to derive all signal timing from

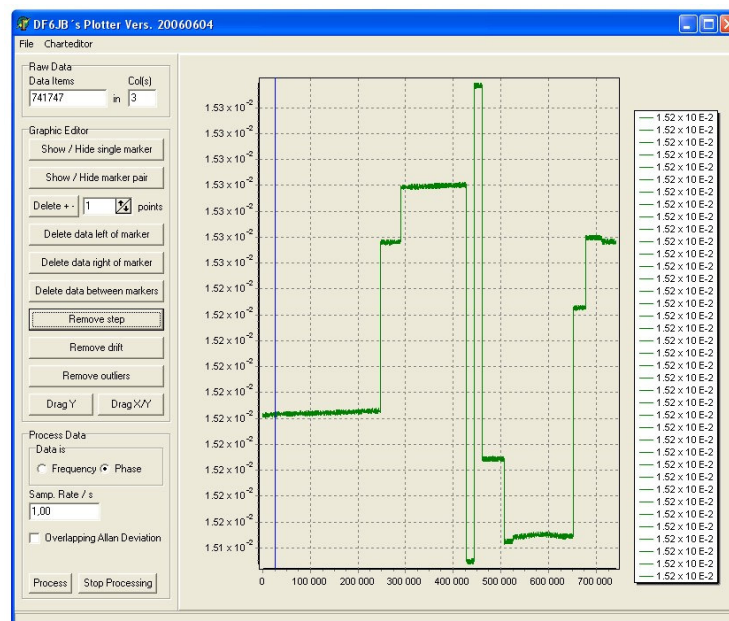
including horizontal a vertical sync pulses for the picture transmission. Since the ZDF uses its own net of ball transmitters all over Germany the sync pulses of ZDF should make a nice and cheap source of stability when building frequency standards.

As the diagram shows there are lots of sudden phase jumps which may or may not originate from switching between different equipment in the transmission central. Between the steps the comparison does not look too bad, so we have to get rid of the steps first before we can see any further.

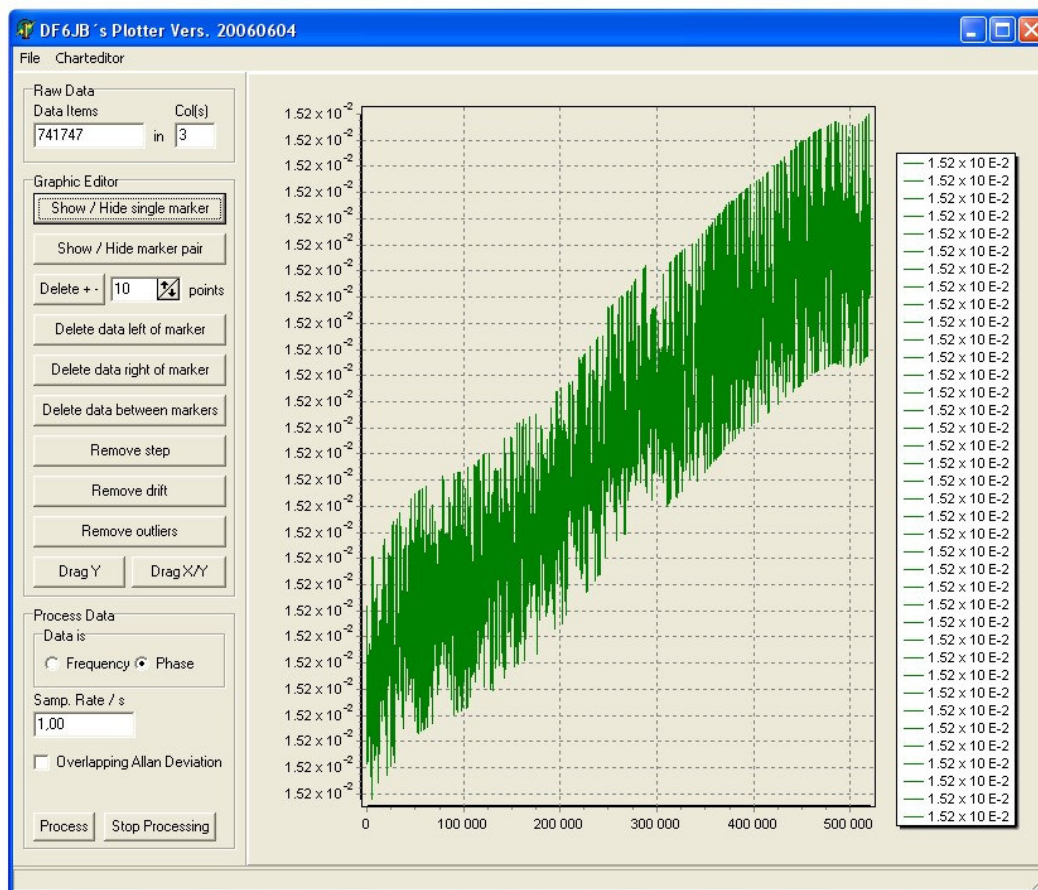
Zoom in the first step until it looks like below and place a single cursor as central as possible over the step:



Press **Remove Step** and un-zoom the window. Voila, the step is no more existent!

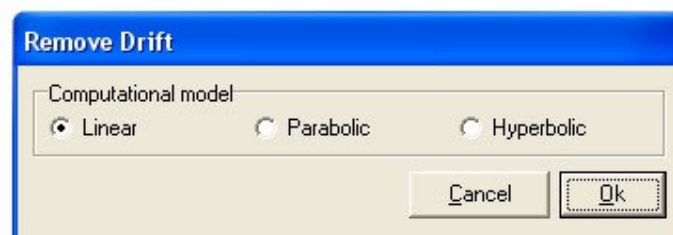


Some mouse clicks later after removing the rest of the steps the display shows

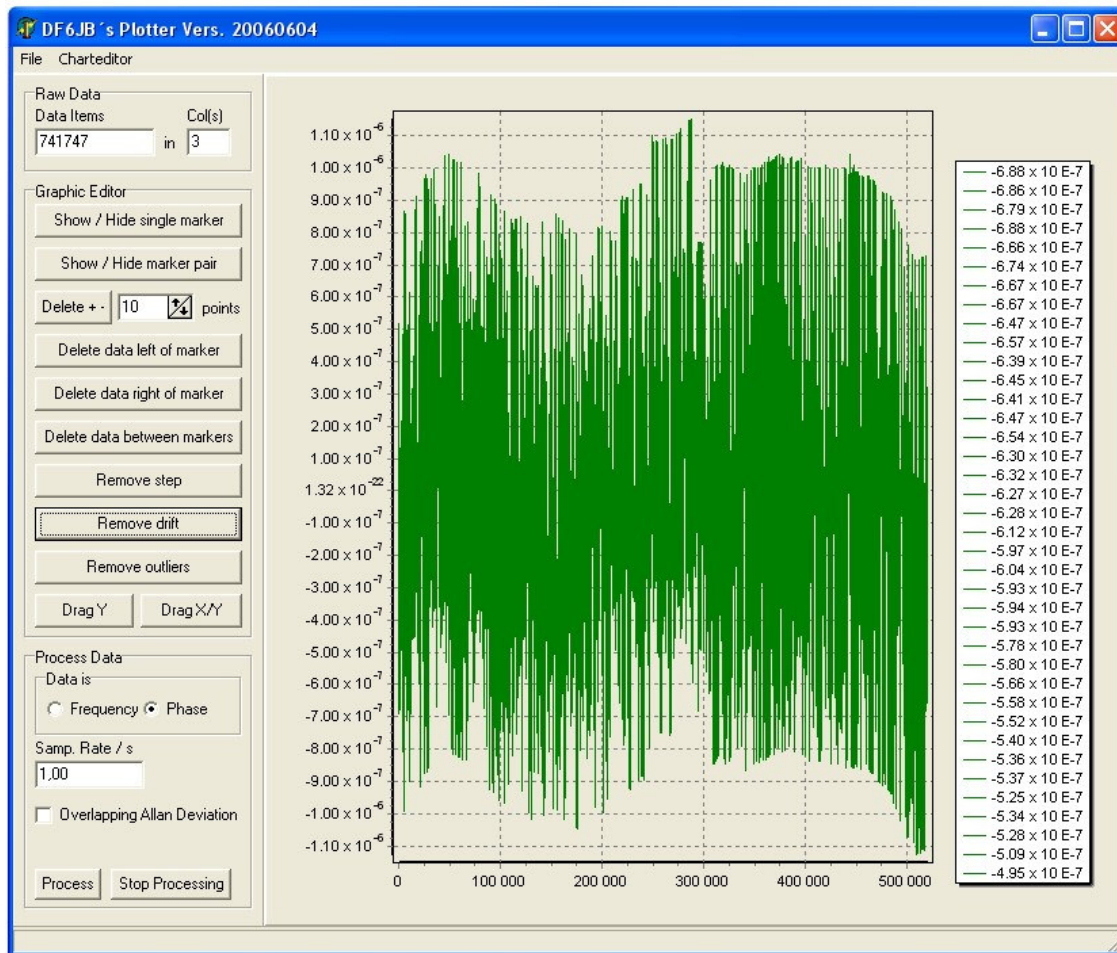


The phase delay between my house standard and ZDF constantly grows. If they have a rubidium standard in Mainz, then what we see is just the result of the rubidium's frequency error. Should their rubidium really be synchronized to PTB what we see is perhaps the result between NIST and PTB differences in standard frequency but I guess the first applies.

Next we shall remove this linear phase drift. Press **Remove drift** and a window opens asking us which kind of drift removal we would like:



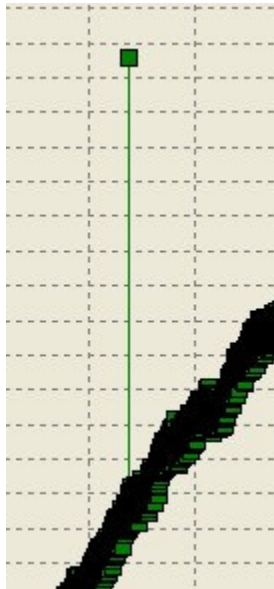
Choose **Linear** and press **Ok** and see the result on the next page.



There are also two non-destructive ways to manipulate data. Consider something like this:



Instead of cutting out this outlier we can select **Drag Y** from the graphic editor which results in a display like



Note that every single data point is now represented by a square marker. If you come close to such a marker with the mouse cursor, the cursor changes to a *hand symbol*. By pressing the left mouse button now and moving the mouse you will notice that you may drag the point's position in vertical direction (only). If you are ready with dragging points press the **Stop button**. Using **Drag X/Y** you may drag data points in both orientations. Note that dragging point uses lots of processing power. The less points are visible in the graphic area the faster and smoother the drag operation will run. For that reason always use the highest zoom level that is applicable for your drag operation.

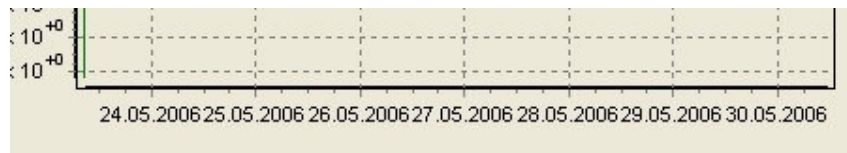
More or less these are the capabilities of the graphic editor. Play around with it to get a feeling for it. Note that **Plotter** will always make a *copy* of the data if you open a file. Your original data will be in a virgin state even if you performed the wildest changes using the graphic editor. All changes are performed in the pc's ram only.

If you come to the conclusion that the result of your changes should be saved to disk use the **File Save as** menu entry. Note that there is no **File Save** menu entry. You will always have to give the file a filename explicitly. You can, however, overwrite your original data by giving your save file the same name as your original data.

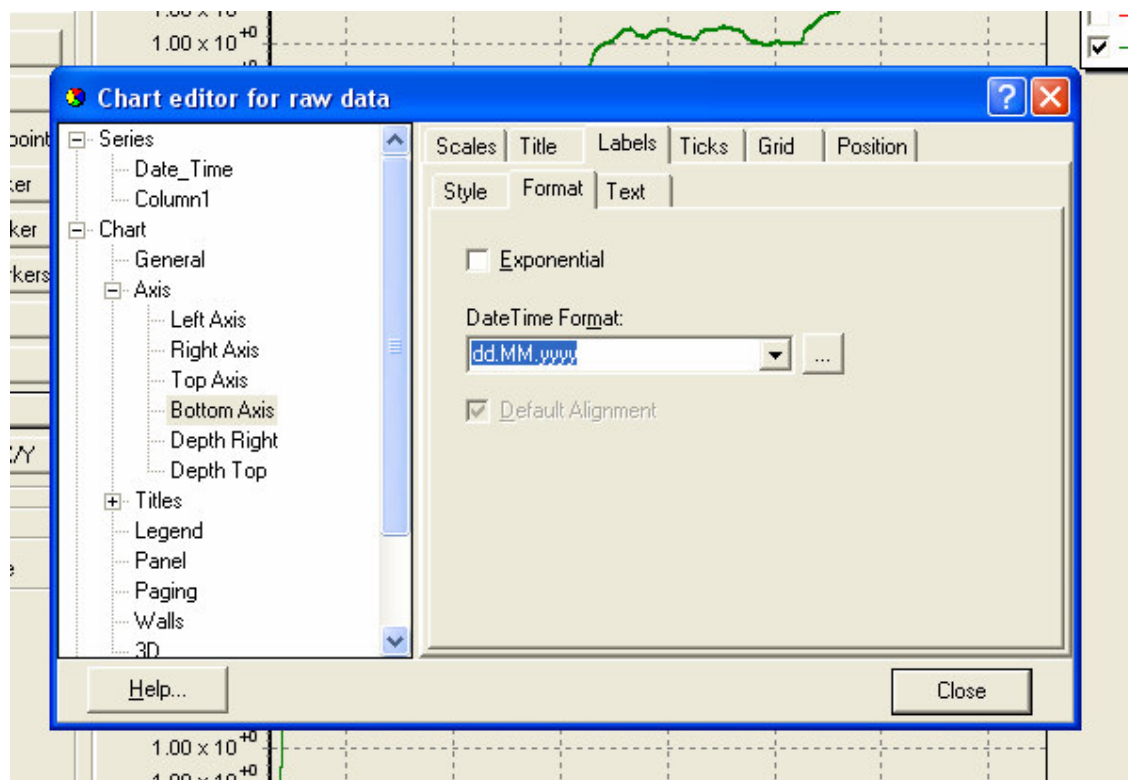
One additional note to graphic editing: Even if none of the data columns is explicitly considered a date/time information **Plotter** will always expect the data has been measured one after another which implies the horizontal axis always has a date/time like character even if it displays simply measurement point numbers. That has an important consequence that you have to be aware of:

If you cut data from a *single data column* this is like cutting a certain piece of time out of *the whole measurement*. Cutting data from a single column will have the effect of cutting the same amount of time from *all data columns*. This is necessary because if no explicit date time stamp is available then the *index* (the point number) of corresponding data points (corresponding in time!) has to stay the same whatever else may happen.

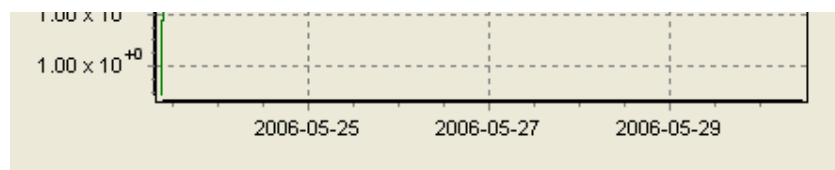
Let's now go to into making a graphic really look professional. We start with the example from a few pages up. First note: Whenever you know that your data contains date time stamps then load the data date time stamped! That will make the horizontal axis self explaining when the data was taken and how long it was taken. In our example, we see that the data is for about six days in May 2006.



You people outside Europe may not really be happy with the notation to be seen here, so let's change it on the fly. Open the **Charteditor**. Go to **Chart, Axis, Bottom Axis, Labels, Format**. What you see is:

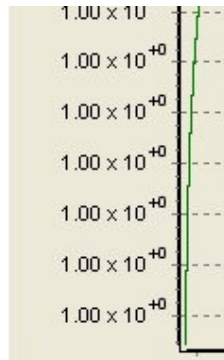


Clearly the **dd.MM.yyyy** setting is what makes the format annoying to you. Change it to **yyyy-mm-dd** which makes the horizontal axis look like

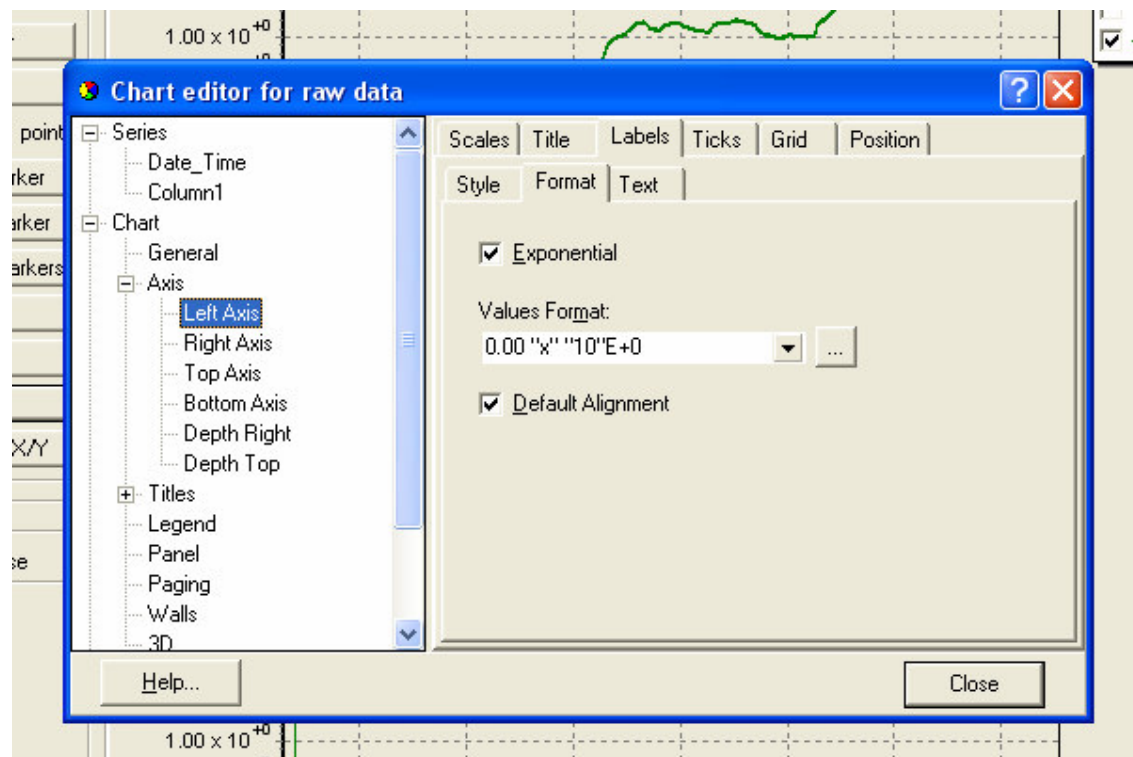


or set the format string to whatever you like.

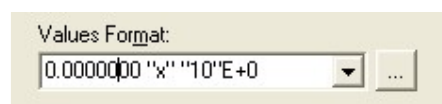
Being at the point of changing **Axis** settings, note that exponential format for the vertical axis is ok but does not show enough digits to make it readable.



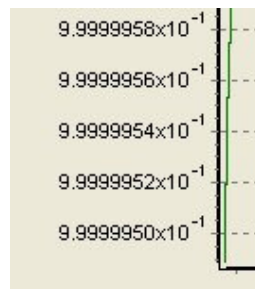
Go to **Chart, Axis, Left Axis, Labels, Format** to see



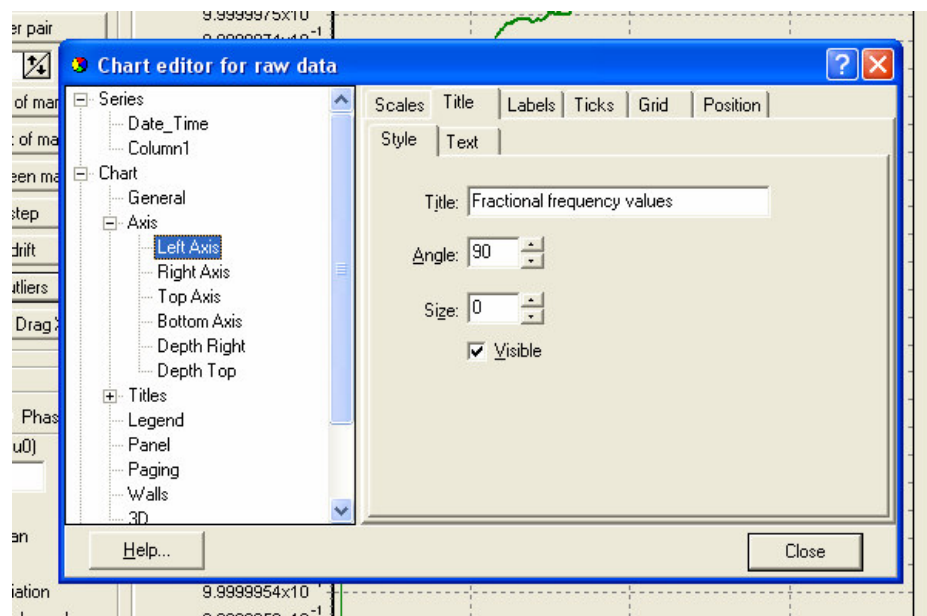
Change the format string to



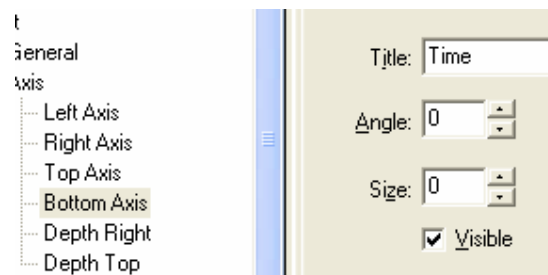
Now the vertical axis displays nicely like:



When giving a graphic to another person it should be absolutely clear, what is displayed in it. This clarity involves saying what the axes display. So add a text “Fractional frequency values” after having gone to **Chart, Axis, Left Axis, Title, Style**.

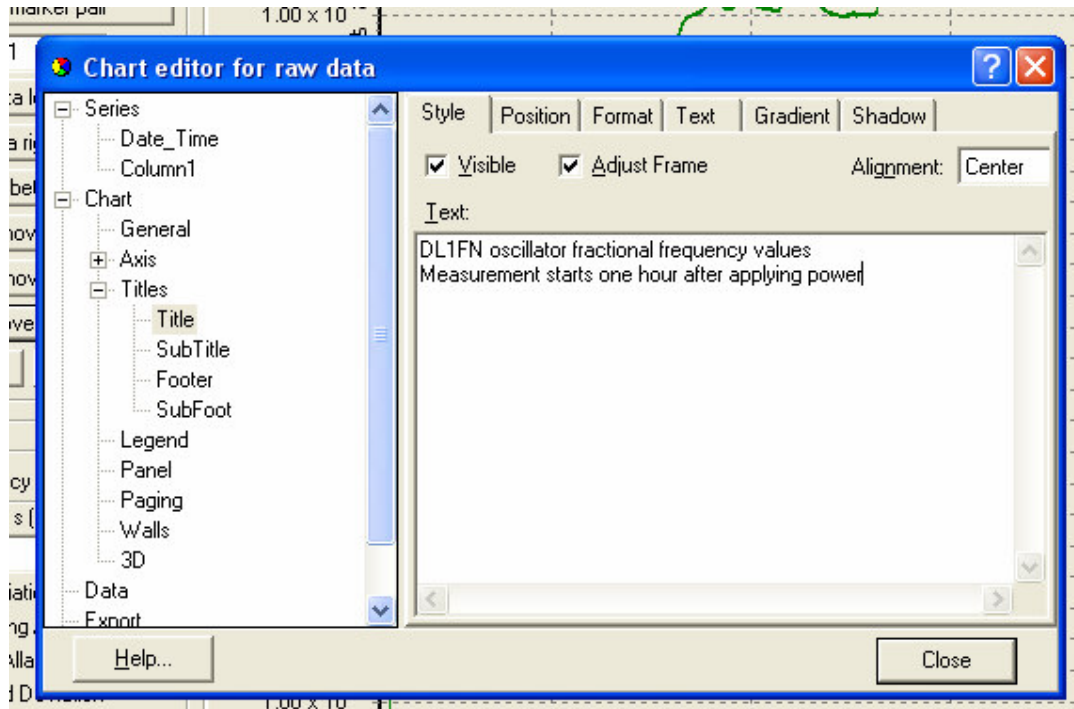


Do the same for the **Bottom Axis**

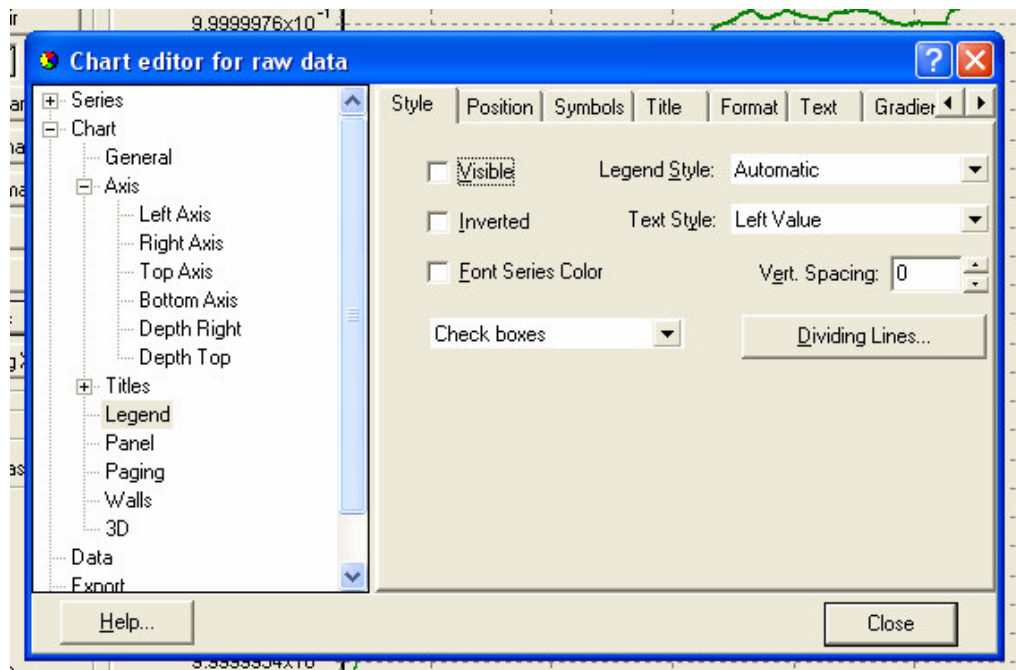


to give the horizontal axis a “Time” title. And yes, the graphics as a whole should get a title too to give the viewer an immediate overview about the facts contained in the graphics.

Go to **Chart, Titles, Title, Style** and enter a title text. Use more than one line if you like.

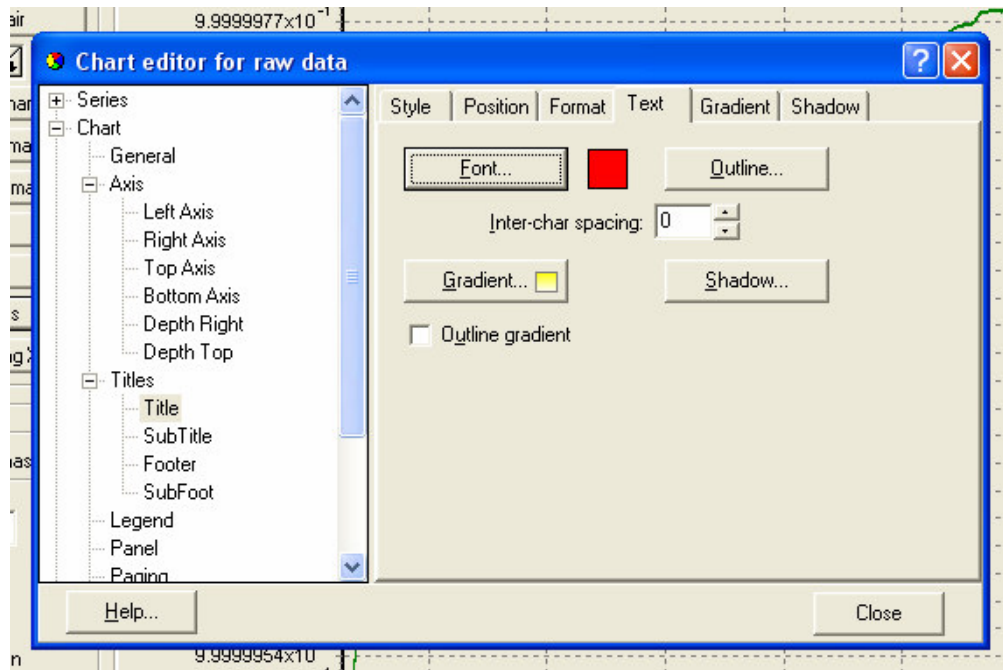


We do not want a legend in this case, so go to **Chart, Legend, Style** and uncheck the **Visible** check box.

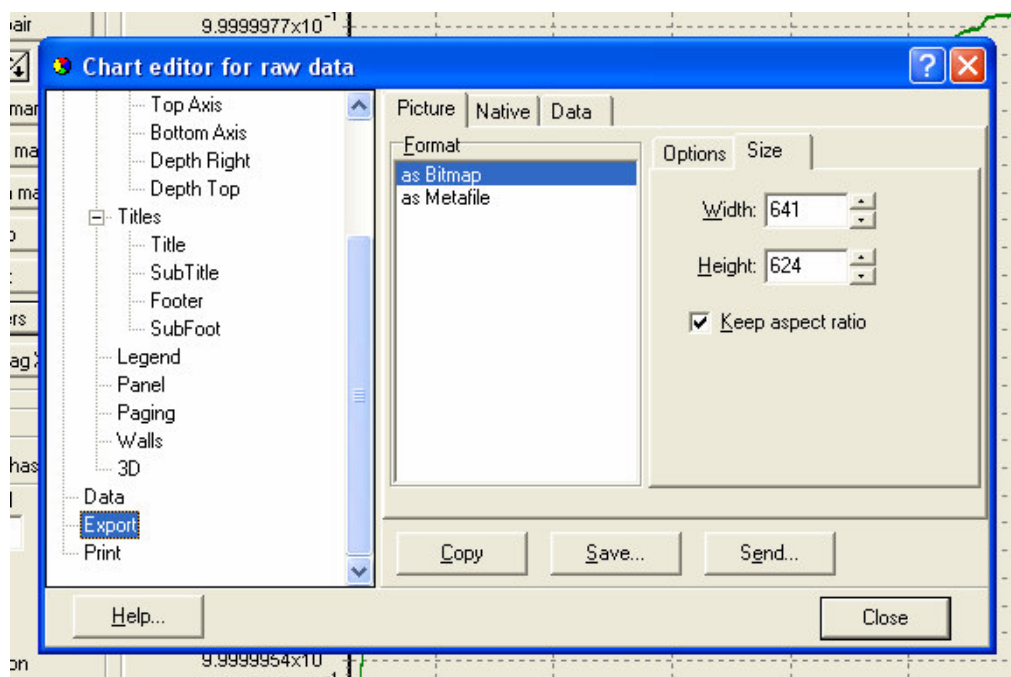


Now the graphics look good to me except the title font could be a bit more bigger and I like it red instead of blue.

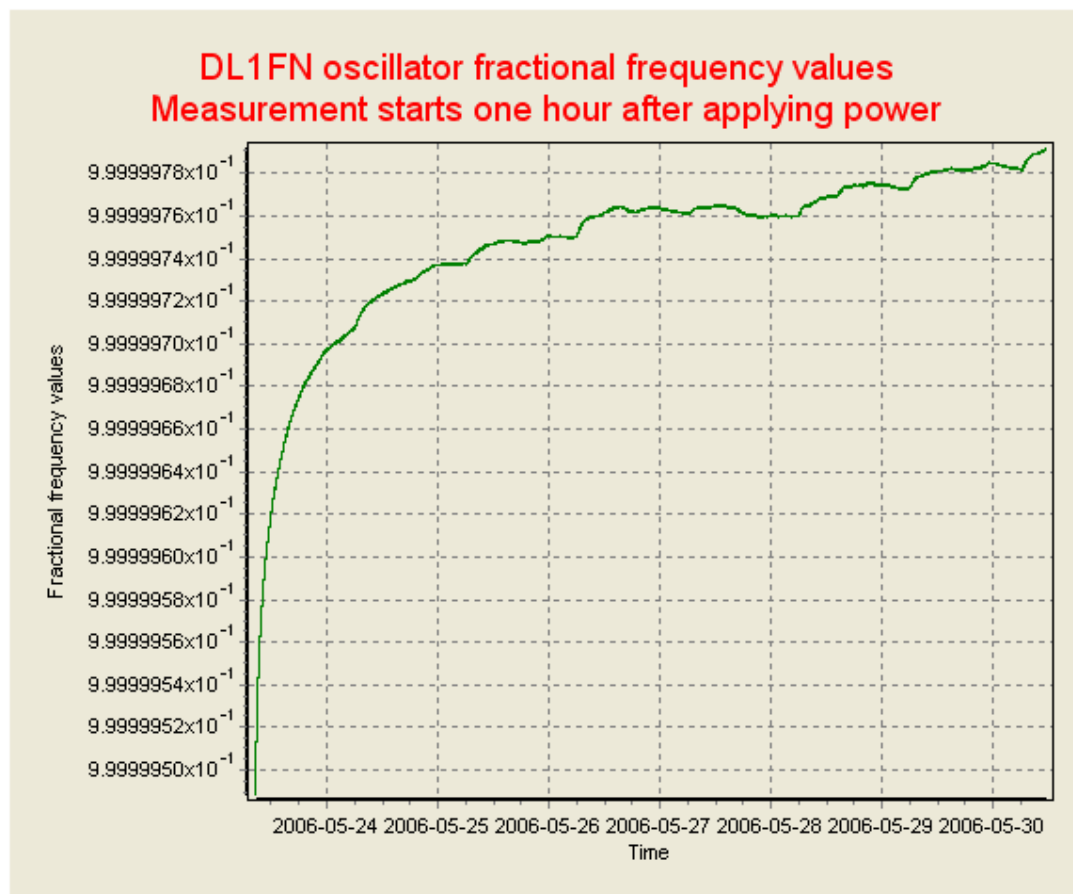
Go to **Chart, Titles, Title, Text** and choose a bigger or even different font and change the text colour to red



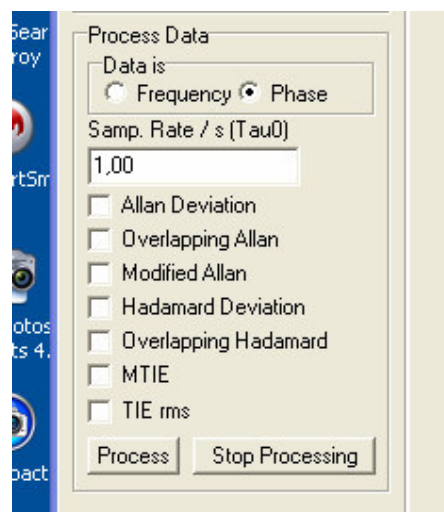
Now that the chart looks ok, export it by going to **Export Picture** and save it as a **bitmap**



Now look at the next graphics. It has been directly imported in this Word document as the bitmap that we have generated in the last step. Looks nice, eeheh?



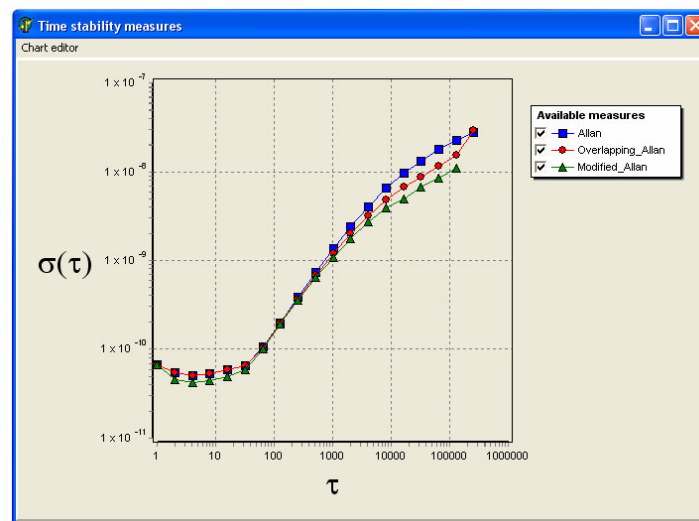
Up to now I showed you how to use **Plotter** to manipulate data and to display data nicely. In addition **Plotter** can be used to compute statistical information for you that is significant in time stability analysis. Currently the following processing capabilities are built in.



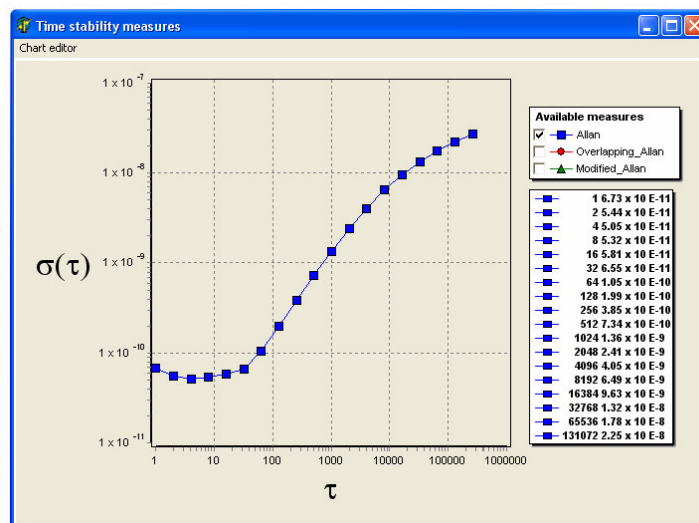
The three Allan deviations are commonly known. Hadamard is not well known, google for it to learn more about it. TIE rms is the root mean square time interval error and MTIE is the maximum time interval error. Note that computing MTIE needs a lot of processing power and may need a lot of time depending on the size of your data.

Before processing the data you need to tell **Plotter** whether it is phase or frequency data. TIE rms and MTIE can only be computed on phase data. You also have to give **Plotter** the information what the sample rate, i.e. the τ_0 of the data is. This applies also for time stamped data!

Once you have done this, simple press Process to open a new window showing the different statistical measures like shown below.



Note, that if you uncheck the legend's checkboxes leaving only one checked, then the data of the checked measure will be displayed.



Note also that this window has a chart editor of its own, so there is a wide range of possible chart manipulations.

