1 Use of M-Files CP42MEA and PLOTMEA

First copy the M-Files into a directory, which is known by MatLab. These M-Files are functions, not script files.

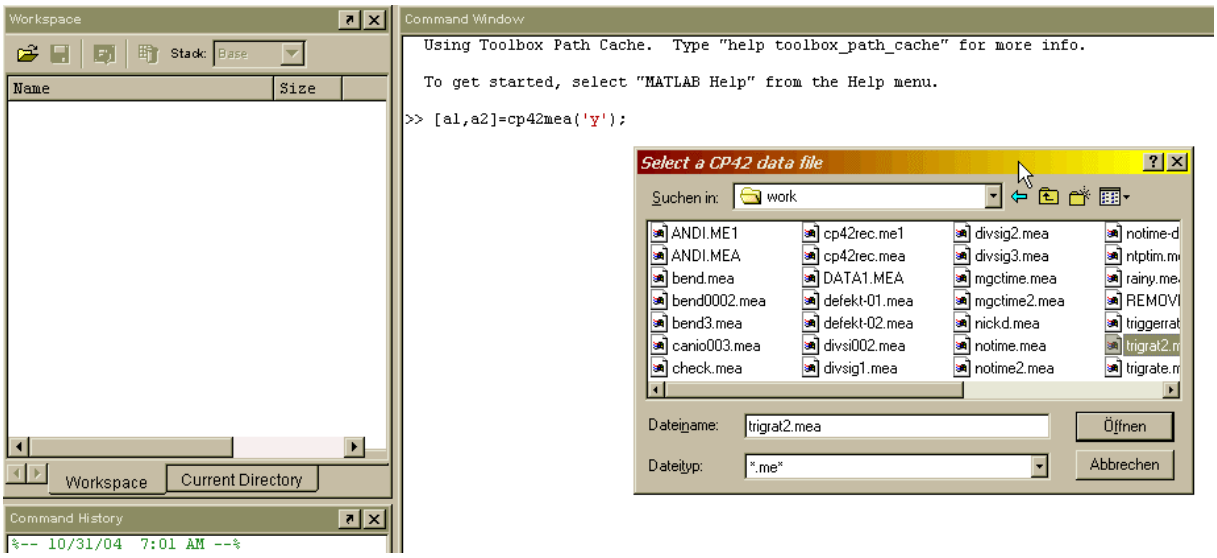
A call like "cp42mea(...);" will not deliver any results. Output arguments are necessary.

```
[a1,a2]=cp42mea('y');
```

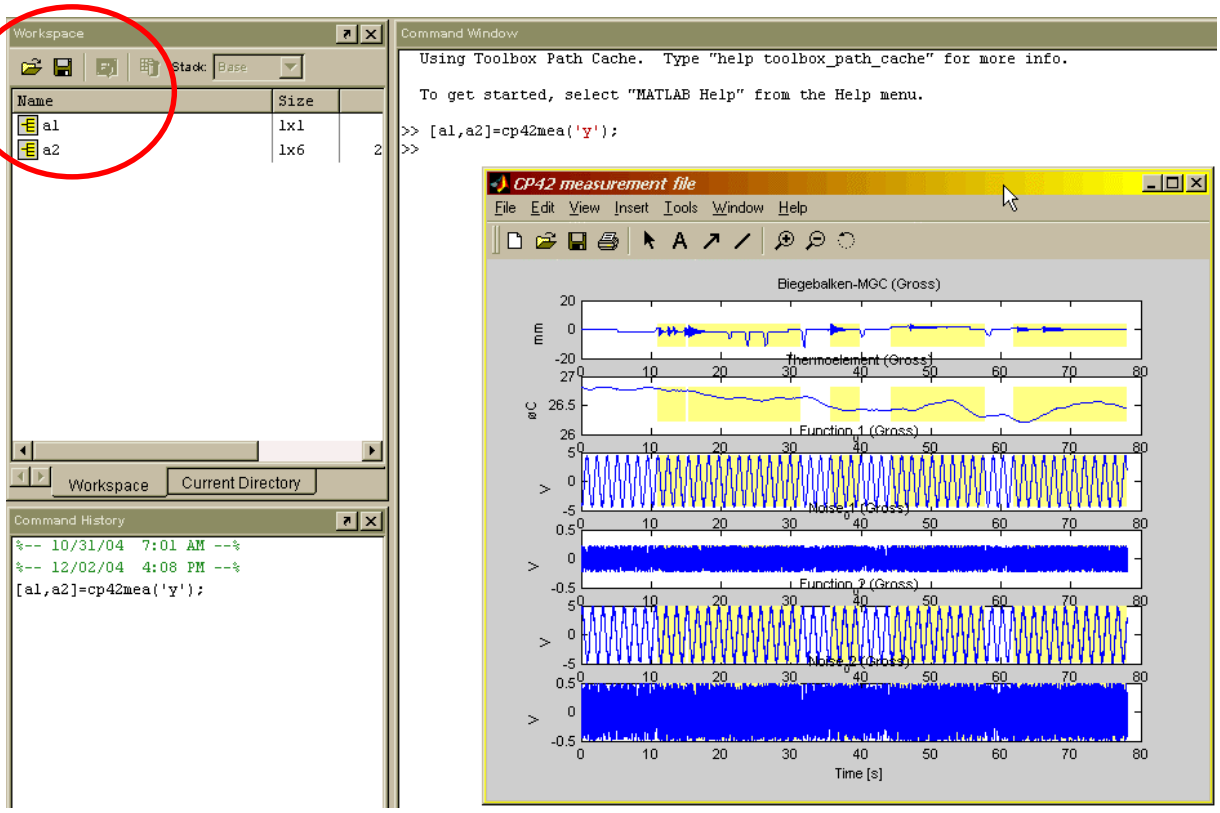
The string 'y' leads to a graphical presentation in a MatLab figure if the amount of channels is not too big. The "File open" dialogue will appear and files with extension "*.me*" can be selected.

Important:
Only MEA-Files with measurement value format 4 Byte INT (Intel) are supported.
Supported Time formats are:

- **No time information**
- **MGCplus device Time (64 Bit)**
- **NTP time channel**



The result of the function are 2 structure variables: a1 contains file information, the variable a2 the measurement information (channel names, units, meas. data and more). In the "Workspace"-windows these variables are listed.



When typing just a1 in the workspace window, the file information is displayed.

```
>> [a1,a2]=cp42mea('y');
>> a1

a1 =

        FileID: 6001
      ByteLine: 32
      MWformat: 1253
    DataOffset: 2080
           cmt: [1x80 char]
      creation: [1x26 char]
      filename: 'C:\MATLAB6p5\work\trigrat2.mea'
      ChannelC: 8
       MVLines: 24000
         MRate: 100
    TimeFormat: 7000
      filesize: 770080
        cmtpar: [1x80 char]

>>
```

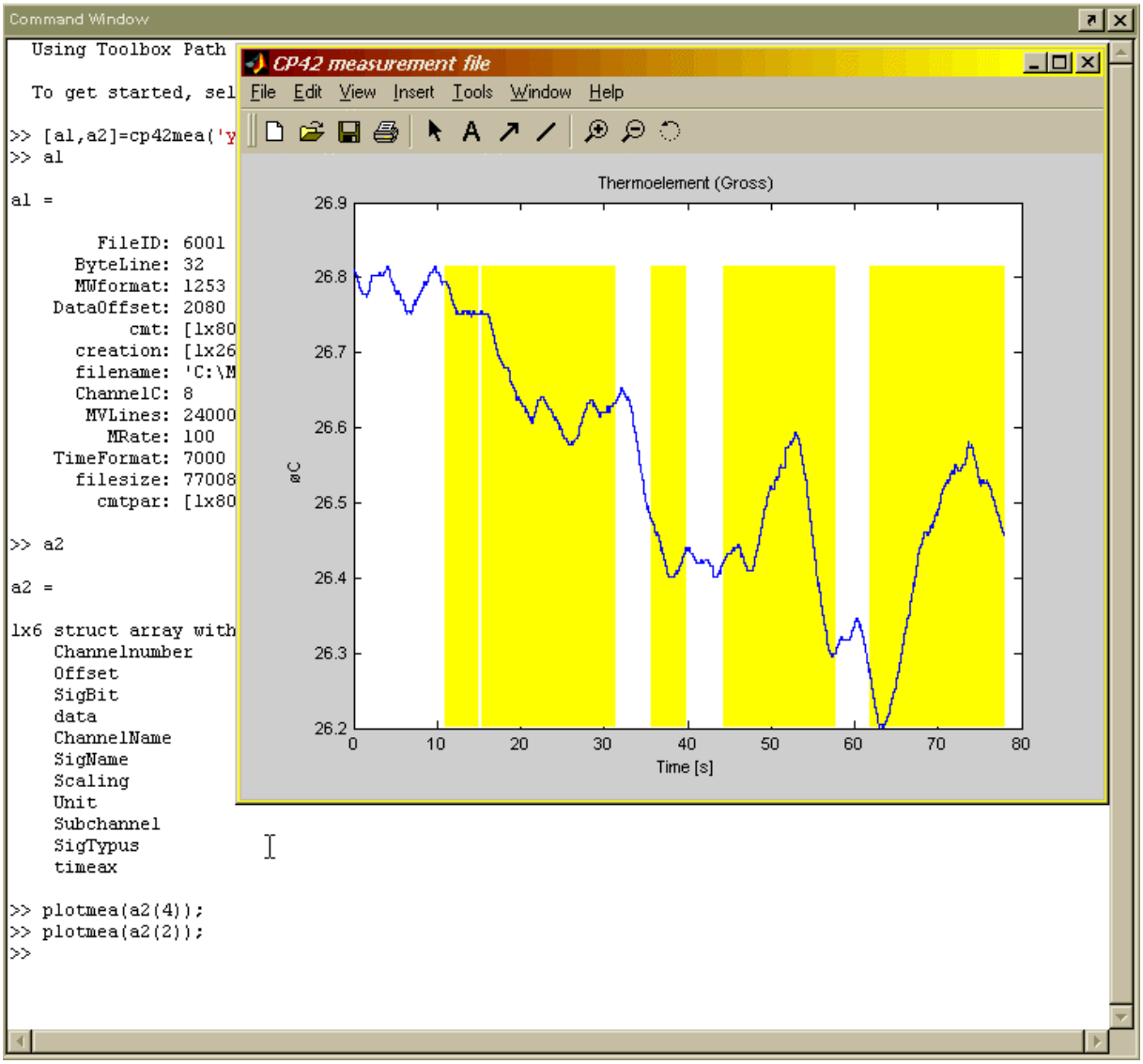
When typing just a2, only the structure can be seen, becuase a2 is an arry of the structure. If a plot of a measurement is desired, the command line below can be used.

```
plotmea(a2(4)); % a2: Structure with measurement data
```

New since version 1.3:
You can use also the commands

```
plotmea(a2(4),1);
plotmea(a2(4),2);
```

to show status information of the measurement values. This status contains info about limit values, overflow, channel error etc.

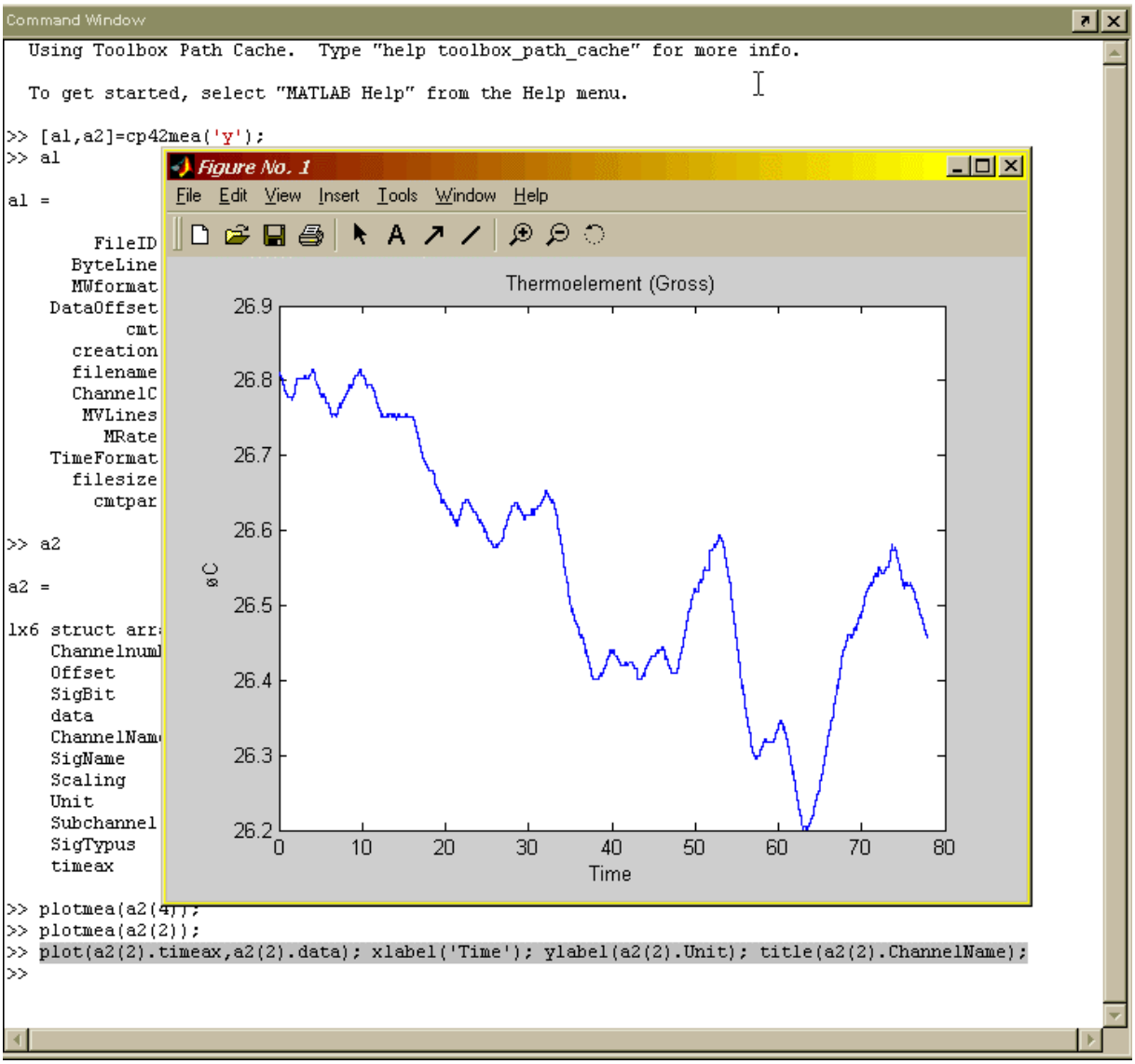


the yellow shaded areas are time ranges, where the input signal was sampled with a higher measurement rate. (Measurement Rate Trigger for MGCplus)

If preferred, own matlab figures can be created (see command lines below):

```
plot(a2(2).timeax,a2(2).data); xlabel('Time');
ylabel(a2(2).Unit); title(a2(2).ChannelName);
```

The result is shown in the screenshot below.



These functionalities are defined as function, so that every one can create ist own dataview or post process evaluation programs.

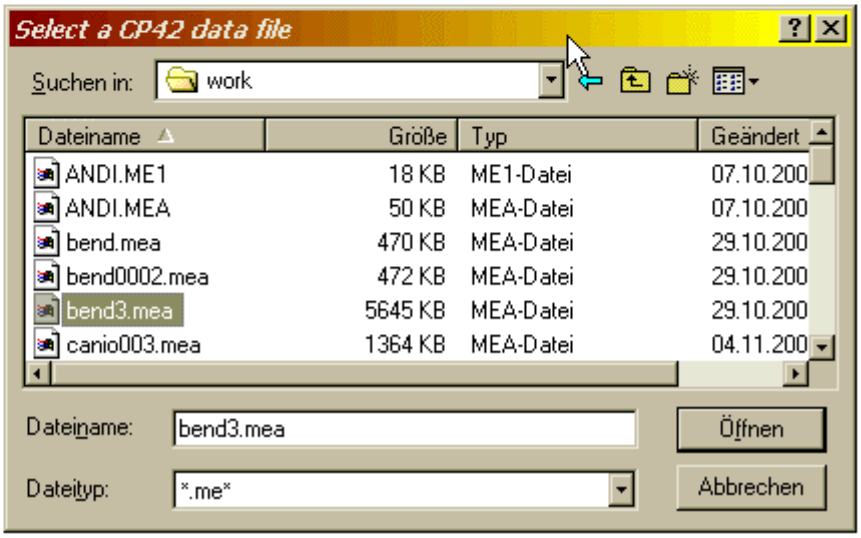
2 Reading large measurement files

Sometimes a large file should be read in only partially. The functions "meacomp.m" and "cp42cut.m" can be used for that task.

At 1st, a large file can be read in "compressed" with the function "meacomp.m". The measurement data of every channel is read in blockwise. From these blocks the minimum and maximum value is evaluated. For each channel and signal an "activity signal" [0%...100%] is derived.

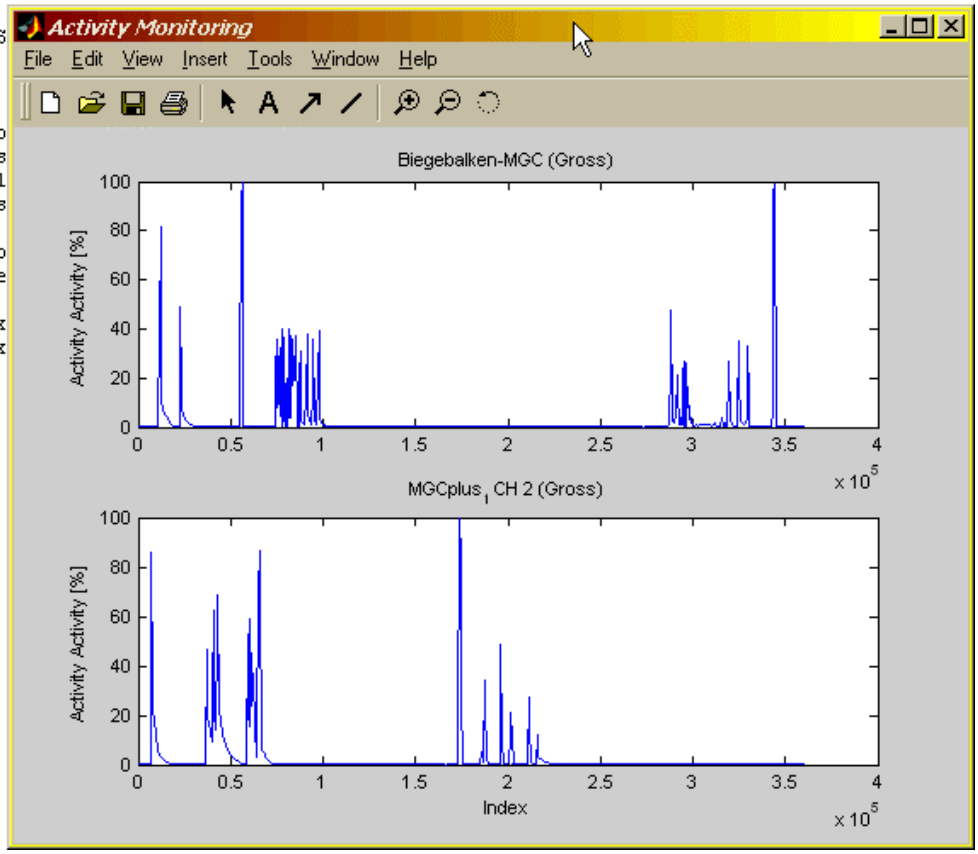
```
>> help meacomp

  MEACOMP
          [FILEINFO,CHINFO]=MEACOMP([MEAFILE],[COMPRESSION])

          MEAFILE:        Name of a CP42 measurement file with
                          Measurement value format 4 Byte INT.
          COMPRESSION:    Samples / Block.

          MEACOMP reads the values of the CP42 measurement file.
          A block containing all channels/signals is read. The
          number of samples is specified by the variable
          compression. From every channel/signal the max. and
          min. value are subtracted, so that you get an information
          about "signal activity" in each block.

          The information about the file is stored in the struct
          variable FILEINFO, the information about channels and
          signals is contained in the variable CHINFO.
          CHINFO(k).data contains the "compressed" activity information,
          CHINFO(k).timeax contains the sample index.
```

```
>> [a1,a2]=meacomp;
```



During the reading process a progress indicator is displayed in an additional matlab figure. After reading a matlab figure with the activity curves appears, if the channel amount is not too high.

```
>> help meacomp

  MEACOMP
          [FILEINFO,CHINFO]=MEACOMP([MEAFILE],[COMPRESSION])

          MEAFILE:        Name of a CP42 measurement file with

          COMPRESS

          MEACOMP
          A block
          number o
          compress
          min. val
          about "s

          The info
          variable
          signals
          CHINFO(k
          CHINFO(k

>> [a1,a2]=meacomp;
>>
```

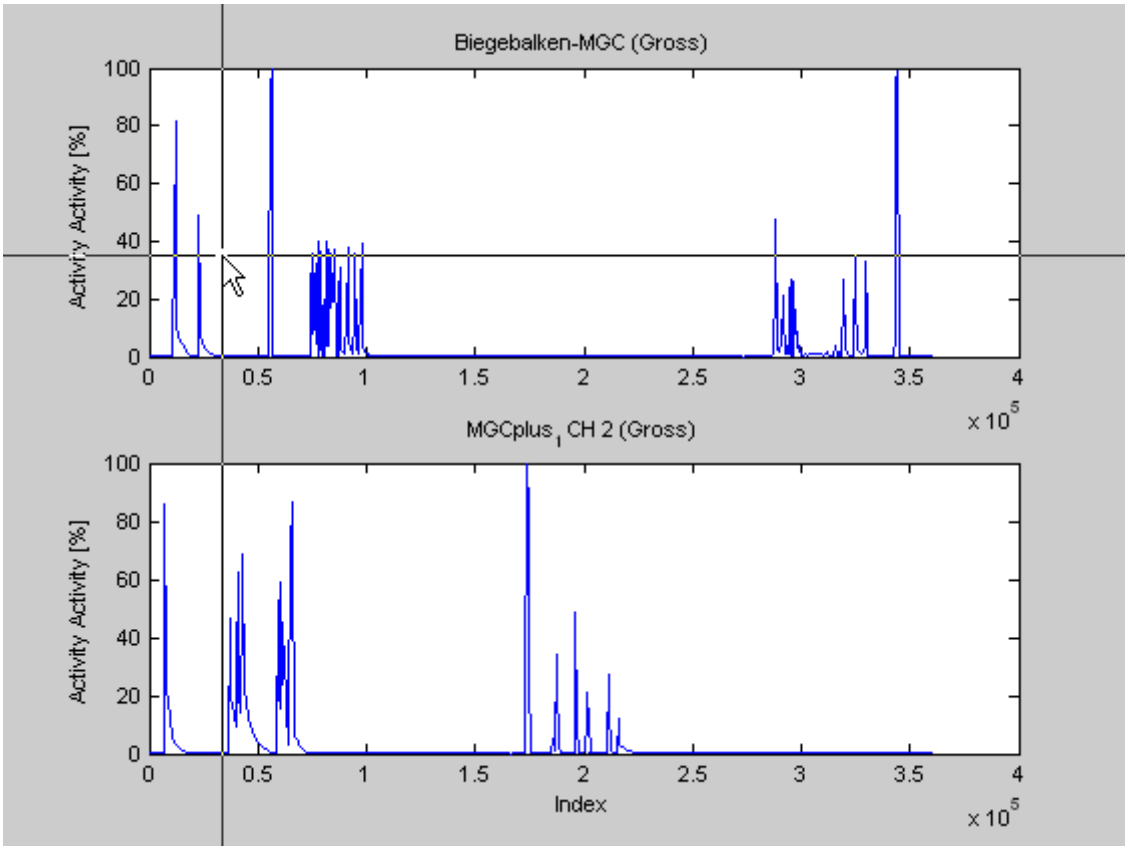


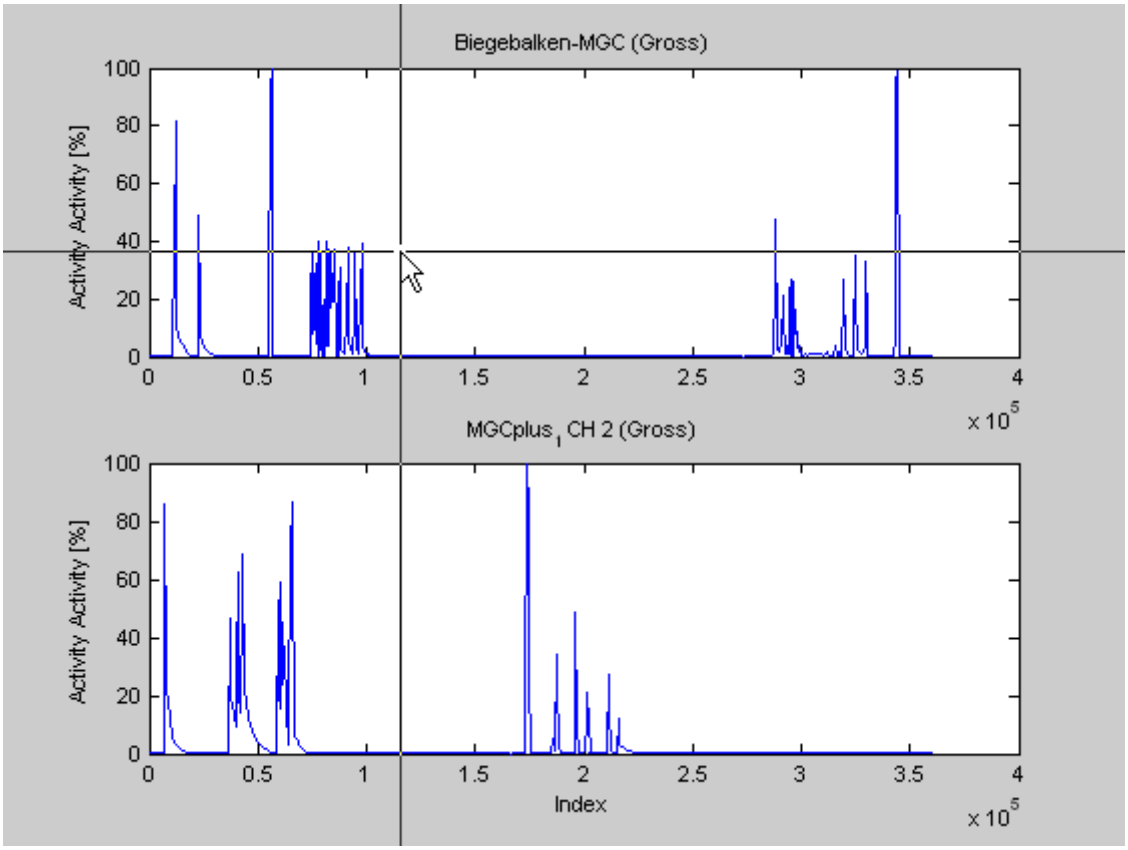
Now a selection of the range of interest is recommended. The matlab standard function "ginput.m" can be used for that.

Important:
The abscissa does not contain time information. It contains the sample index.

Below the use of "ginput.m" is shown. The next page shows as an example the performed selection.

```
>> [x,y]=ginput(2);
>> x

x =

  1.0e+005 *

    0.3088
    1.1475
```

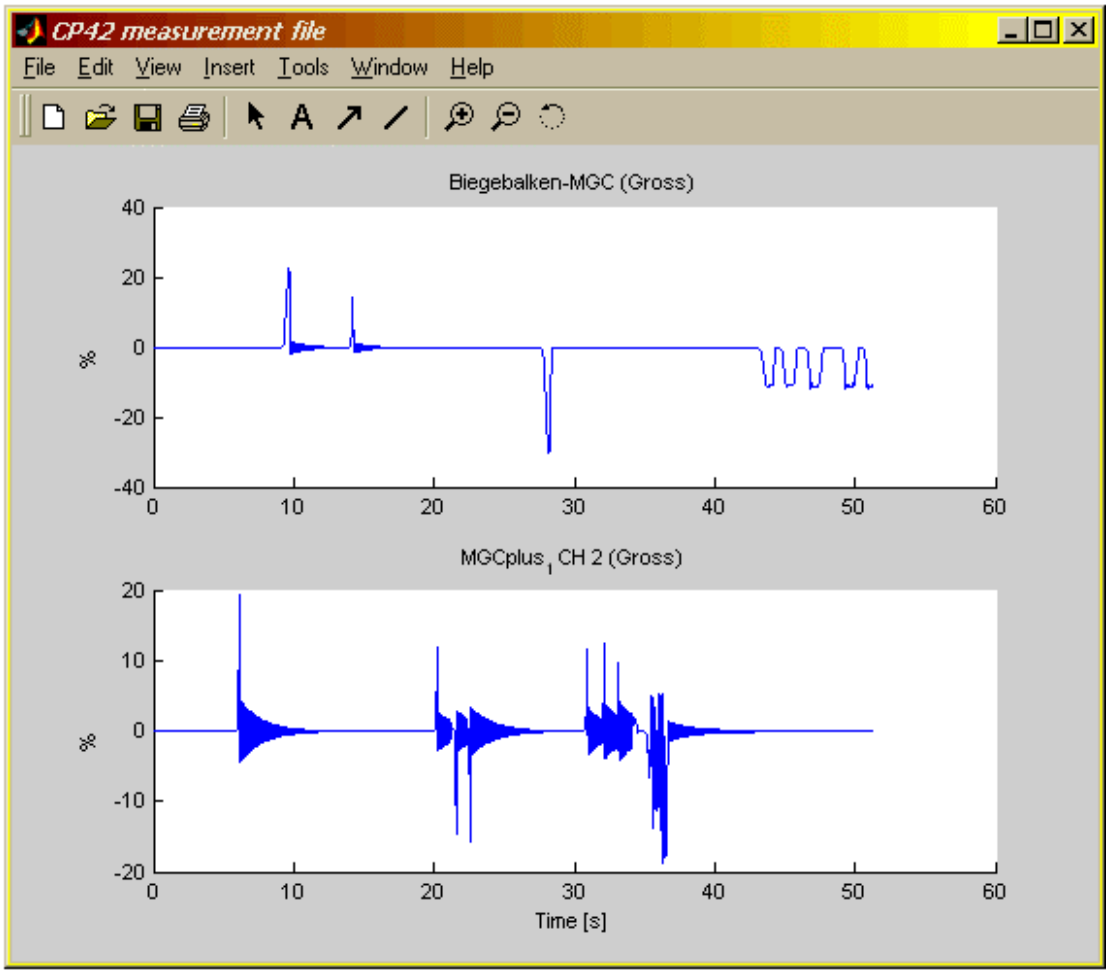

Biegebalken-MGC (Gross)
Activity Activity [%]
MGCplus_1 CH 2 (Gross)
Index
x 10^5


Biegebalken-MGC (Gross)
Activity Activity [%]
MGCplus_1 CH 2 (Gross)
Index
x 10^5

The 2-element columnvector x contains the selected sample index range. The desired selective reading of the large measurement file is performed by the matlab function "cp42cut.m".

```
>> help cp42cut

  cp42cut   Read MGCplus Data files partly, recorded with CP42

            [fileinfo,CHINFO]=cp42cut([meafile],[idx])

            meafile is the complete filename of the CP42 DAQ file.
            idx is a vecot of length 2, containing the start- and
            end index of the samples to be read in.
            Both input arguments are optional.

            This function reads the data files recorded on a PC Card
            harddisk on CP42. The input argument "meafile" is optional,
            when it is missing, a user dialog for selecting a file with
            extension *.me* is shown.

            fileinfo is a struct, containing the FileID, the number of
            channels "ChannelC", Byte/Line, the data format, the data offset,
            the time format, the file comment, the comment of the current
            parameterset, the information about the creation date and the
            number of used slots.

            CHINFO is a struct, containing information about the channel
            number, the scaling, offset, unit, signalling bit, subchannel,
            data, timeaxis, channel name and the channel code.

            With PLOT(CHINFO(k).timeax,CHINFO(k).data) a measurement
            signal can be displayed.
            TITLE(CHINFO(k).ChannelName) adds the channel name as
            graph title
            YLABEL(CHINFO(k).Unit) puts the used unit on the Y axis.

            There is a lot of information in the mea files about channels
            and slots. Not all of these information is evaluated.
            Want more ? Have a look at the word file "cp42cut-FORMAT.DOC"

            Only CP42 measurement files are supported. These DAQ files may
            contain time information in following formats:
            1) No Time Information
            2) MGCplus Time (64 Bit)
            3) NTP Time Format
            The Time Formats "USB Frame Count" and 32 Bit MGC device Time
            are not supported.

            IMPORTANT:
            Only Measurement files with format 4 BYTE INT (Intel) are
            supported.

>> [b1,b2]=cp42cut(a1.filename,x.');
```

As result the file information b1 and measuremend data information b2 is returned. The measurement data information contains correct time information.




```
-> [b1,b2]=cp42cut(a1.filename,x.');
-> plotmea(b2);
```

Using the matlab function plotmea shows the data.

3 The Measurement Value Status

A measurement value from MGCplus contains of 32 Bit. 24 Bit contain information about the measurement value itself (24 Bit resolution of AD converter). 8 Bits contain information about the status of the measurement value.
The status contain information about limit value switches, overflow, calibration error and a change flag.

- **Bit 0: Status Limit Value 1**
- **Bit 1: Status Limit Value 2**
- **Bit 2: Status Limit Value 3**
- **Bit 3: Status Limit Value 4**
- **Bit 4: Overflow Gross Value**
- **Bit 5: Overflow Net Value**
- **Bit 6: calibration error**
- **Bit 7: Change-Flag (amplifier scaling has changed)**

When using for example the function

```
[a1,a2]=CP42MEA;
```

then a2.status contains as a decimal value the state of the binary status bits. The measurement signal and its status can be displayed very easy with the function plotmea, which has been extended.

When using

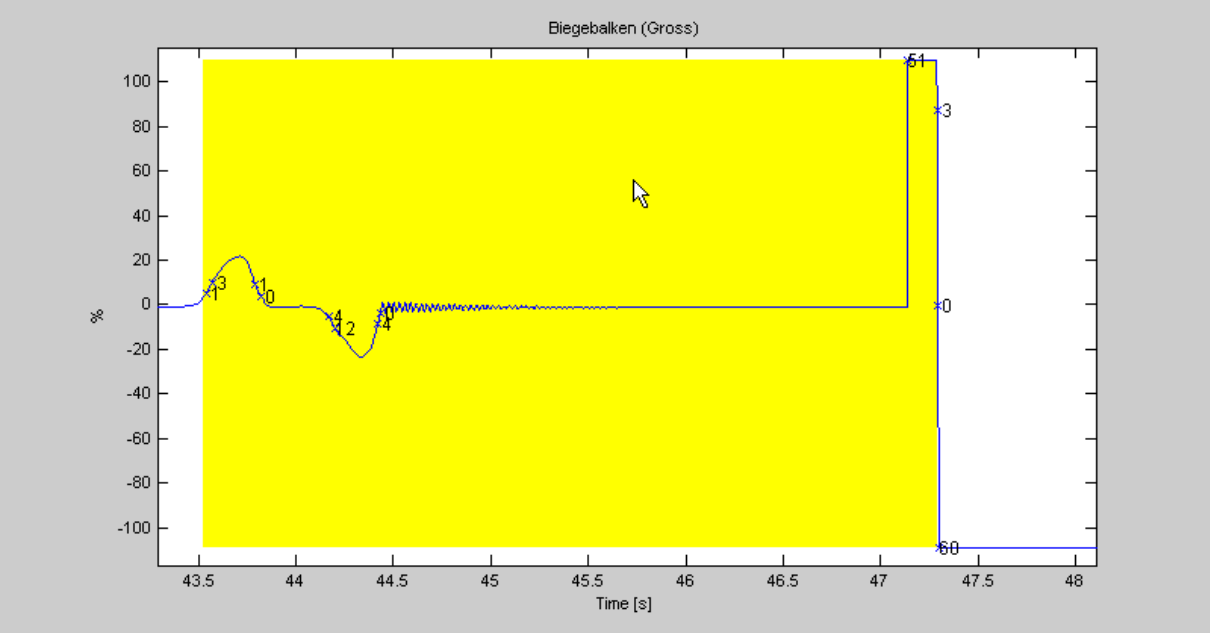
```
plotmea(a2,1);
```

the measurement signal is shown together with markers, which indicate the position, where the status has been changed.

With the function

```
plotmea(a2,2);
```

the markers are additionaly labelled with the decimal representation of the status information.



In the 1st part, the cange of limit value switches can be seen, the 2nd part shows the effect of disconnecting a half bridge transducer. The status 51 and 60 indicate a overflow of the gross and net value. (Only the state of limit values has been changed)

To extract all the changes of the measurement status, you can use a new function called meastatus.m. The typical use case is:

```
[a1,a2]=cp42mea;
S=meastatus(a2);
```

S is a structured variable, S.status contains the changes of the measurement status. S.status is a matrix with 4 rows: the 1st 2 rows contain the time and meas.-value, where the status has a change, row 3 contains the running index and the 4th row the new status.