An Axiometrix Solutions Brand

Exercises for imc FAMOS II – Digital Course

- Block 4 - Doc. Rev.: 1.2- 28.08.2025

Exercise A

Exercise objective:

In addition to writing complete measurement datasets to Excel files, it is also possible to run Excel remotely from FAMOS. In this exercise, an evaluation of all measurements in a folder is to be performed and the results summarized and transferred to an Excel file.

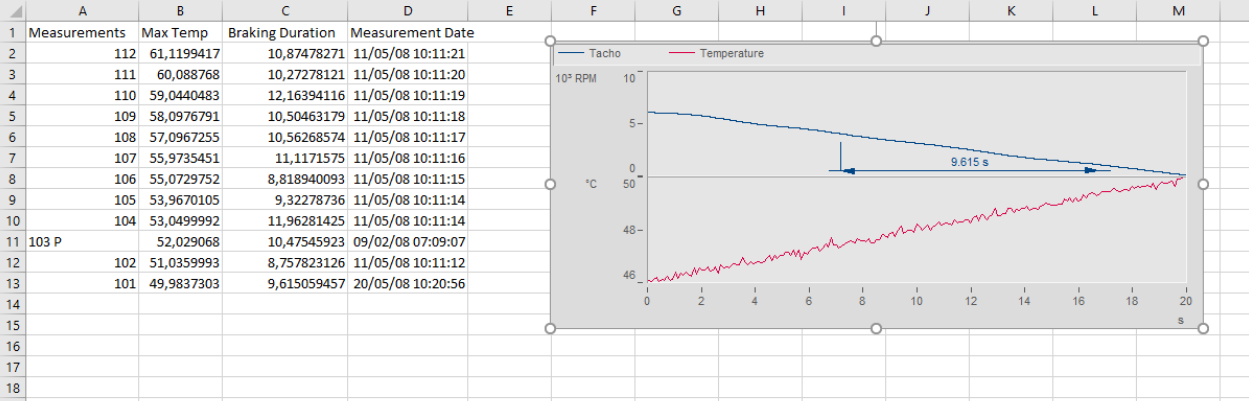
Task formulation:

Determine the maximum temperature and the duration of braking from 4000 to 1000 RPM for all measurements in the **Test** folder and place these in an Excel list. In addition, the creation time of the channel **Tacho** as well as the directory name should also be transferred and placed in the Excel list.

Additional task:
Add a curve window with the channels **Tacho** and **Temperatur** as well as a time difference marker between the two threshold speed values used for the braking time evaluation to the Excel spreadsheet.

Result:

You will receive an Excel worksheet with the desired entries and an image of the generated curve window. The resulting Excel worksheet looks something like the following:



	A	B	C	D
1	Measurements	Max Temp	Braking Duration	Measurement Date
2	112	61,1199417	10,87478271	11/05/08 10:11:21
3	111	60,088768	10,27278121	11/05/08 10:11:20
4	110	59,0440483	12,16394116	11/05/08 10:11:19
5	109	58,0976791	10,50463179	11/05/08 10:11:18
6	108	57,0967255	10,56268574	11/05/08 10:11:17
7	107	55,9735451	11,1171575	11/05/08 10:11:16
8	106	55,0729752	8,818940093	11/05/08 10:11:15
9	105	53,9670105	9,32278736	11/05/08 10:11:14
10	104	53,0499992	11,96281425	11/05/08 10:11:14
11	103 P	52,029068	10,47545923	09/02/08 07:09:07
12	102	51,0359993	8,757823126	11/05/08 10:11:12
13	101	49,9837303	9,615059457	20/05/08 10:20:56
14				
15				
16				
17				
18				

Exercise steps:

- Open the group of commands for remote control of Excel in the Function Library. You will find it under **Files, Import/Export** in the **Excel (Extended)** subgroup. Create a new sequence where you first start Excel with suitable commands, make it visible and create a new workbook.

  ```
  err = XlStart()
  XlVisible(1)
  XlWbNew("")
  ```

- Create a new text array containing the titles of the columns (Measurements, Max Temp, Braking Duration, Measurement Date).

  ```
  titles = ["Measurements", "Max Temp", "Braking Duration",
  "Measurement Date"]
  ```

- Set the headings in the Excel spreadsheet using a loop and suitable functions from the Excel kit:

```
For i = 1 To TxArrayGetSize(titles)
   reference = XlBuildA1Ref(1,i,1,1)
   XlSetText(reference, titles[i])
end
```

- Load all **.raw** data from the subdirectories in the Test folder from the sample data one after the other. Use exercise C from block 3 as a guide.
  Tip: Instead of listing all files in the subdirectories first and then loading them in a loop one after the other, the **FileLoad()** function can be used instead to load all files at once. A measurement reference can be omitted in this case. Since the further processing is still carried out in the same loop, the corresponding channels can be overwritten again during the next loop pass.

```
list_of_dirs = FsFileListNew(basepath ,"*",1, 0, 2)
number_of_dirs = FsFileListGetCount(list_of_dirs)
For i = 1 to number_of_dirs
   act_dir = FsFileListGetName(list_of_dirs, i)
   FileLoad(act_dir + "\*.raw","",0)
   ; further code
End
FsFileListClose(list_of_dirs)
```

- Calculate the maximum of the **Temperature** channel as well as the braking time from 4000 RPM to 1000 RPM. Also determine the trigger time of **Tacho** as well as the current measurement directory:

```
maxTemp = Max(Temperature)
braking_duration = Pos(Tacho, 1000) - Pos(Tacho,4000)
meas_date = Time?(Tacho)
measurement = FsSplitPath(act_dir, 2)
```

- Transfer the values to the Excel table using suitable functions from the Excel kit. In doing so, consider the respective variable type:

```
reference = XlBuildA1Ref(i+1,1,1,1)
XlSetText(reference, measurement)
reference = XlBuildA1Ref(i+1,2,1,1)
XlSetValue(reference, maxTemp,0)
reference = XlBuildA1Ref(i+1,3,1,1)
XlSetValue(reference, braking_duration,0)
reference = XlBuildA1Ref(i+1,4,1,1)
XlSetValue(reference, meas_date,1)
```

- After processing the loop, set the format for the column **Measurement date**:

```
reference = XlBuildA1Ref(2,4, number_of_dirs,1)
XlSetCellFormat(reference, "mm/tt/jj hh:mm:ss")
```

- Save the Excel file in the **basepath** and assign a suitable file name:

```
XlsName = basepath + "\summary.xlsx"
XlWbSave(XlsName,0)
```

- Close the opened Excel file at the end of the sequence and save it.

```
XlWbClose()
```

Additional task: Transfer of a curve window to an Excel file as picture element.

- Create a new curve window and configure it as you wish. If you want to create a curve window by sequence, you will find an example code block at the end of this exercise.
- Copy the curve window using the CwAction() function and then paste it into the Excel spreadsheet. Of course, these commands must be executed before closing the Excel file:

```
CwAction("clipboard.copy")
XlPaste("f2")
```

Code block for creating a suitable curve window for the exercise. The explanation of the individual steps is not part of the exercise, so this block remains uncommented at this point:

```
CwNewWindow(Tacho,"show")
CwNewChannel("append last axis", Tacho)
CwNewChannel("append new cosys", Temperature )
CwSelectByIndex("cosys", 1)
CwNewElement("marker.abs")
CwMarkerSet("x", Posi(Tacho,1000))
CwMarkerSet("y", 1000)
CwMarkerSet("arrow", 0)
CwMarkerSet("text", "")
CwMarkerSet("linelength", 0)
CwNewElement("marker.abs")
CwMarkerSet("x", Posi(Tacho,4000))
CwMarkerSet("y", 4000)
CwMarkerSet("arrow", 0)
CwMarkerSet("text", "")
CwMarkerSet("linelength", 0)
CwNewElement("marker.hori.dimline")
CwMarkerSet("ref.1", 2)
CwMarkerSet("ref.2", 1)
```

Note: You can find a complete solution sequence for this exercise in the sample data (**excel_transfer.seq**). In this sequence a group gr is used to hide variables in the variable list that are not directly necessary or to define them later as local variables with the command line **Local gr** as well.

Exercise B

Exercise objective:

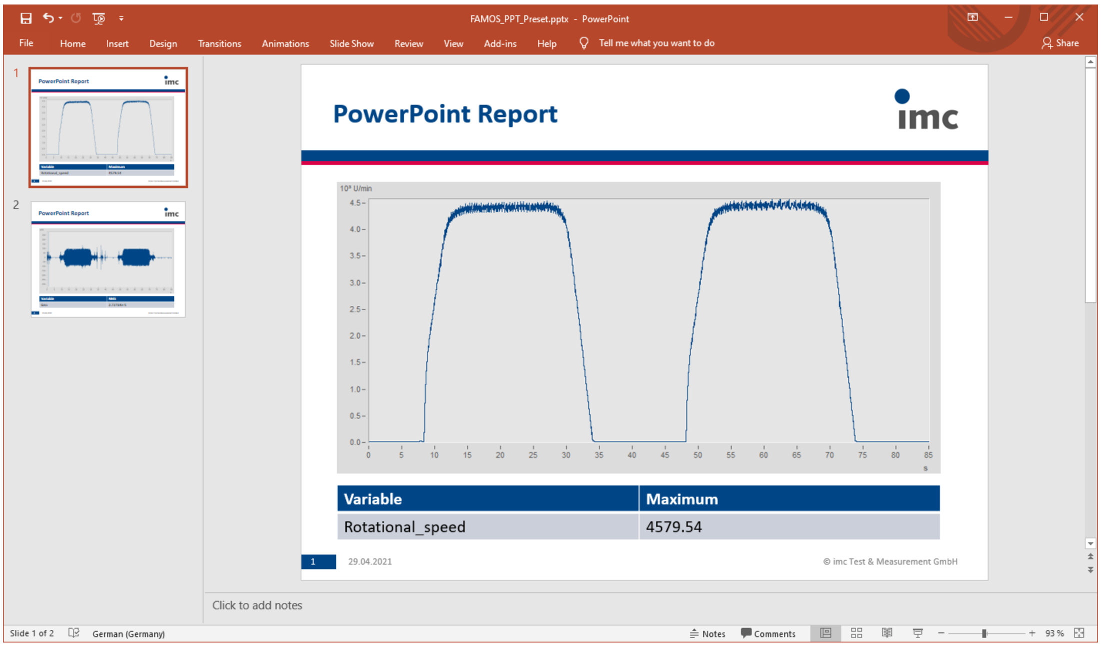
In this exercise you will learn how to use the FAMOS PowerPoint kit to remotely control Microsoft PowerPoint and transfer content by sequence into a PowerPoint presentation.

Task formulation:

Load the first folder from the sample data **Serial_Measurement\2020-11-06 08-54-21 (1)** and fill a PowerPoint (**FAMOS_PPT_Preset.pptx** from the sample data) as a report with a representation of the dataset **Rotational_speed** on the first slide, as well as its calculated maximum as a numerical value in the table below. Then delete the existing slide number 2 and insert the first slide again. Fill it afterwards with the data set **Geo** and its RMS value as numerical value.

Result:

You will receive a PowerPoint presentation with 2 slides, each containing a curve window and a table with results.



PowerPoint Report

imc

Variable	Maximum
Rotational_speed	4579.54

1 29.04.2021 © imc Test & Measurement GmbH

Click to add notes

Exercise steps:

To be able to control PowerPoint remotely from FAMOS, a template presentation must first be created. The individual elements on a PowerPoint slide can be addressed with the PowerPoint Kit via their **alternative text** property. In this exercise the prepared template **FAMOS_PPT_Preset.pptx** from the example files is used.

- Open the template **FAMOS_PPT_Preset.pptx** from the sample files in PowerPoint. This contains a slide with a title text field and a graphic as placeholders. Open the edit bar for the alternative text, find out and note the assigned values for the two placeholders.
- Close PowerPoint and switch to FAMOS.

- Load all files from the first directory from the Serial Measurement sample data **Serial_Measurement\2020-11-06 08-54-21 (1)**:

  ```
  FileLoad(basepath + "\Serial_Measurement\
      2020-11-06 08-54-21 (1)\*.raw","",0)
  ```

  basepath is a variable that contains the path to the sample files on your PC.
- The PowerPoint kit can be found in the Function Library in the **Presentation** group. Switch to the input window and open the template with the command:

  ```
  res = PptOpenPresentation(basepath + "\FAMOS_PPT_Preset.pptx",1)
  ```

  Upon successful execution, PowerPoint should have opened.
- Search for the index of the slide containing the element with the alternative text **CurveWindow**:

  ```
  index = PptFindSlideByAlternativeText(1,"CurveWindow")
  ```

  Since the opened template contains only one slide, the result here is obviously 1 for the index.
- Replace the text of the text box with the alternative text **Headline** on the previously found slide with the text **PowerPoint Report**:

  ```
  PptSetText(index,"Headline","PowerPoint Report")
  ```

- Open the **Rotational_speed** channel in a curve window. Then set the options to select the current curve window and replace the graphic on the PowerPoint slide with an image of the selected curve window.:

  ```
  Show Rotational_speed
  CwSelectMode("newest")
  CwSelectWindow("")
  PptSetCurve(index,"CurveWindow",1,0,"screen")
  ```

- Calculate the maximum of **Rotational_speed** and fill the table on the PowerPoint slide with the corresponding values:

  ```
  ;Calculate maximum and enter in table of presentation
  maximum = Max(Rotational_speed)
  res = PptSetCellText(index,"Table",2,1,"Rotational_speed")
  res = PptSetCellText(index,"Table",1,2,"Maximum")
  res = PptSetCellText(index,"Table",2,2,TForm(maximum,""))
  ```

  The curve window can be closed again afterwards.
- Delete the second slide of the presentation:

  ```
  res = PptDeleteSlide(index + 1)
  ```

- Insert a new slide by appending the first slide of the template once again to the already opened presentation:

  ```
  res = PptAddSlides(index,basepath + "\FAMOS_PPT_Preset.pptx",1,1)
  ```

- Replace the title text of the newly added slide again with **PowerPoint Report**:

  ```
  index = index +1
  PptSetText(index,"Headline","PowerPoint Report")
  ```

- Show the channel **Geo** in a curve window, calculate its RMS value with the function **RMS()** and paste the results on the 2nd page of the PowerPoint presentation. Use the previous steps as a guide:

```
Show Geo
CwSelectMode("newest")
CwSelectWindow("")
PptSetCurve(index,"CurveWindow",1,0,"screen")
rms_value = RMS(Geo)
res = PptSetCellText(index,"Table",2,1,"Geo")
res = PptSetCellText(index,"Table",1,2,"RMS")
res = PptSetCellText(index,"Table",2,2,TForm(rms_value,""))
```

- Save the created presentation under a new name (e.g. Report.pptx):

```
res = PptSavePresentation(basepath + "\Report.pptx")
```

- Close the presentation:

```
res = PptClosePresentation()
```