Perfection in Automation
www.br-automation.com

Library for condition monitoring

AS 4.5

Version 1.0
Date: 22/10/2019

Contents

1 Introduction

The library “mod_ConMon” simplifies condition monitoring on B&R PLCs. The sample can be used to evaluate and monitor a machine without knowing the damage frequency beforehand. The machine is recorded in “good” condition and then continuously monitored against the reference data. An alarm is triggered if one of the reference value exceeds a certain limit.

The sample project also includes a task “CM_RawData” that can be used to upload the raw data from the module.

All sample code in this documentation uses Structure Text and marked with a gray frame. The recommended task class is #8. When using the buffer upload it is necessary to run the task at the same speed as the bus where the module is connected.

System requirements
This library was developed and tested with Automation Studio 4.5

- PLC OS system B4.45 or higher
- Intel, Atom target (i386, SG4)
- mappAlarm, mappRecipe, mappView Version 5.7.1 or higher
- X20CM4810 module

2 Function call

The library contains only one function call that has one parameter.

```
fctConMon(ptrConMon := ADR(CM_Analyse))
```

The parameter is a variable of type “CM_ANALYSE_typ”. This structure contains all variables to control and parametrize the condition-monitoring sample. Remember this variable is a pointer. This means that in IEC languages “ADR” must be added.

3 Parameter structures

3.1 Main structure

The condition monitoring function uses a parameter structure to operate the library and transfer information. This structure is split into 3 sub structures.

CMD		Command structure to trigger an action.
	BackupConfig	The condition monitoring function manipulates parameters of the condition monitoring module. To save the existing parameters made in Automation Studio use this command to back up the current parameters.
	RestoreConfig	Use this command to restore the settings previously saved with backup up command.
	Analyse	Start analysing the data from the module. Use 'PAR.mode' to control the type of anaylsis.
	Continue	Acknolwedge current data set and aquire new data for next frequency.
	ErrorReset	Reset pending errors.
PAR		
	RecordName	Variable name where the current data is stored.
	FileDevice	This is where the data is stored. See appendix for details how to create a file device. The data is stored in 'C:\ConMon' if empty.
	MpRecipe	mappRecipe link from the configuration in the configuration view
	MpAlarmXCore	mappAlarmX link from the configuration in the configuration view
	MpAlarmXHistory	mappAlarmXHistory link from the configuration in the configuration view
	ModuleName	This is the module position in Automation Studio physical view
	ModuleChannel	IO channel where the sensor is connected on the module (1-4)
	FrequencyResolution	The steps between two measurements (Default: 1)
	FrequencyStart	Starting frequency (Default: 1)
	FrequencyStop	Maximum frequency for the analysing process
	RmsLimitInc	Factor by which the RMS value is allowed to increase. (Default: 2.0)
	RmsLimitLower	RMS values below this limit will be ignored and the factor RmsLimitInc does not apply to these values.
	RmsLimitUpper	Upper limit for RMS. When the value exceeds the upper limit an alarm is triggered even if RmsLimitInc is not exceeded.
	EnvLimitInc	Factor by which the envelope value is allowed to increase. (Default: 2.0)
	EnvLimitLower	Envelope values below this limit will be ignored and the factor EnvLimitInc does not apply to these values.
	EnvLimitUpper	Upper limit for envelope. When the value exceeds the upper limit an alarm is triggered even if EnvLimitInc is not exceeded.
	TimeToStable	When switching to a new frequency the RMS and ENV values need some time to stabelize. This is the time in seconds to wait before the next frequency is checked.
	Mode	Analysing mode Free run: Just run from start to stop frequency Referencing: Run from start to stop frequency and store reference data. Evaluating: Run from start to stop frequency and compare data with reference data.
STA		
	State	Shows the current state in text format
	Status	Shows the current status as integer
	FrequencyValue	Current frequency that is referenced or evaluated
	FrequencyErrorCount	Number of frequencies that exeeded the reference value
	FrequencyErrorLast	Last frequency that that exeeded the reference value
	DataIsValid	Indicates that current data package stored in "RecordName" is valid.

3.2 Record structure

The record structure holds the data that belongs to the current frequency window.

RmsValue	Current RMS acceleration value
RmsReference	The RMS value that was recorded during referencing
RmsExceededCnt	This is the number of times the RMS value exceeded the reference value
EnvValue	Current ENV acceleration value
EnvReference	The ENV value that was recorded during referencing
EnvExceededCnt	This is the number of times the ENV value exceeded the reference value
AlarmDisabled	Suppress alarms when set to true.

4 Description

The library “mod_ConMon” simplifies condition monitoring on B&R PLCs. The library scans a frequency band between “CM_Analyse.PAR.FrequencyStart” and “CM_Analyse.PAR.FrequencyStop”. Since the condition monitoring module is designed to evaluate a speific frequency a window is generated that moves between the start and stop frequency. The size of the window is defined by “CM_Analyse.PAR.FrequencyResolution”. The default window size is 1 Hz.

The data for the current frequency window is stored in the variable defined in “CM_Analyse.PAR.RecordName”. The variable “CM_Analyse.STA.DataIsValid” indicates that a new record set is available. By setting the variable “CM_Analyse.CMD.Continue” the library continues and aquires the data for the next frequency.

The library supports 3 different types of modes.

4.1 Free run

In free run mode the library runs from start to the stop frequency without recording any data. The data is displayed in the variable specified in “CM_Analyse.PAR.RecordName”.

4.2 Referencing

This mode is executed when the machine is in a “good state”. In referencing mode the library records the data and saves it to a CSV file on PLC flash memory. The destination can be specified as file device in “CM_Analyse.PAR.FileDevice”. When no file device is specified the library store the data in directory “C:\ConMon”.

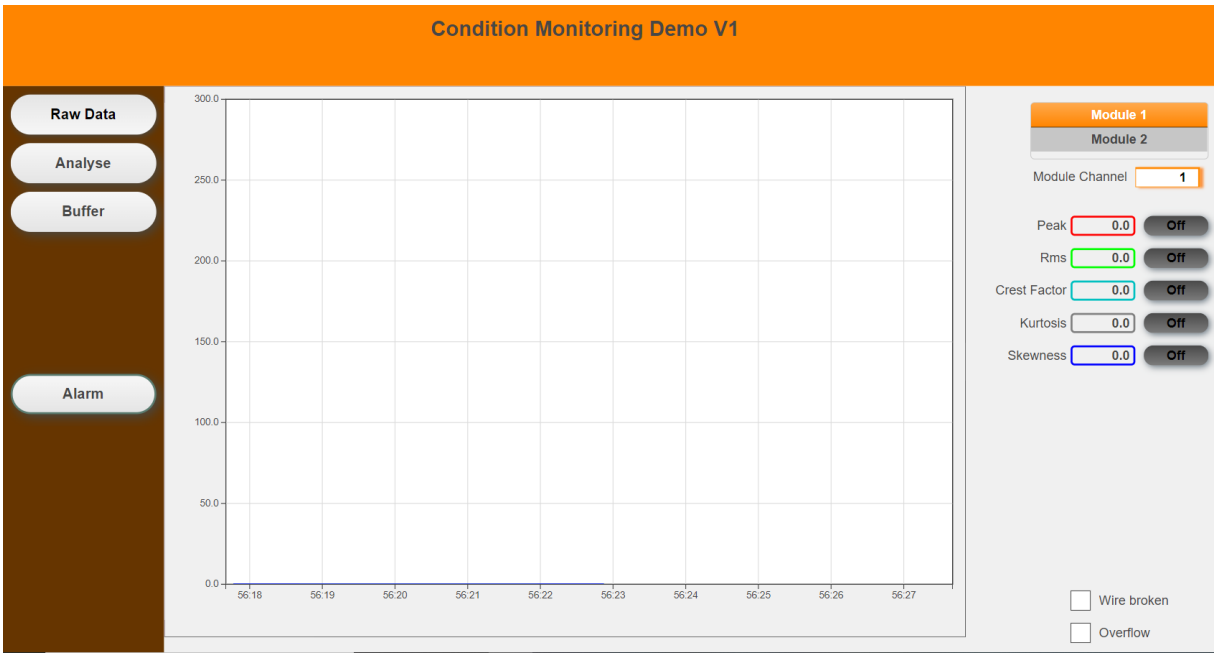
4.3 Evaluating

This mode is executed while the machine is in the field. In this mode the library compares the actual data against the reference data. The general idea is that when the current value exceeds a certain percentage of the referrence value an error is generated. The library uses three criterias to determine if an error must be generated:

- The current value must exceed “CM_Analyse.PAR.RmsLimitInc”. The default value is 100% meaning that the current value must double the reference value to trigger an error.
- To avoid falls triggering from small values the current value must at least exceed “CM_Analyse.PAR.RmsLimitLower”.
- If the current value exceeds “CM_Analyse.PAR.RmsLimitUpper” an error is generated inidepended from the percentage increase.

5 Demo application

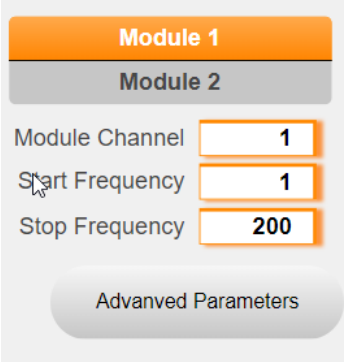
The demo application shows a complete sample for the library in combination with a mappView visualization. The “Raw Data” page shows the current data for the different channels and modules.



This view is used to check if the sensor is producing data and is responding to acceleration.

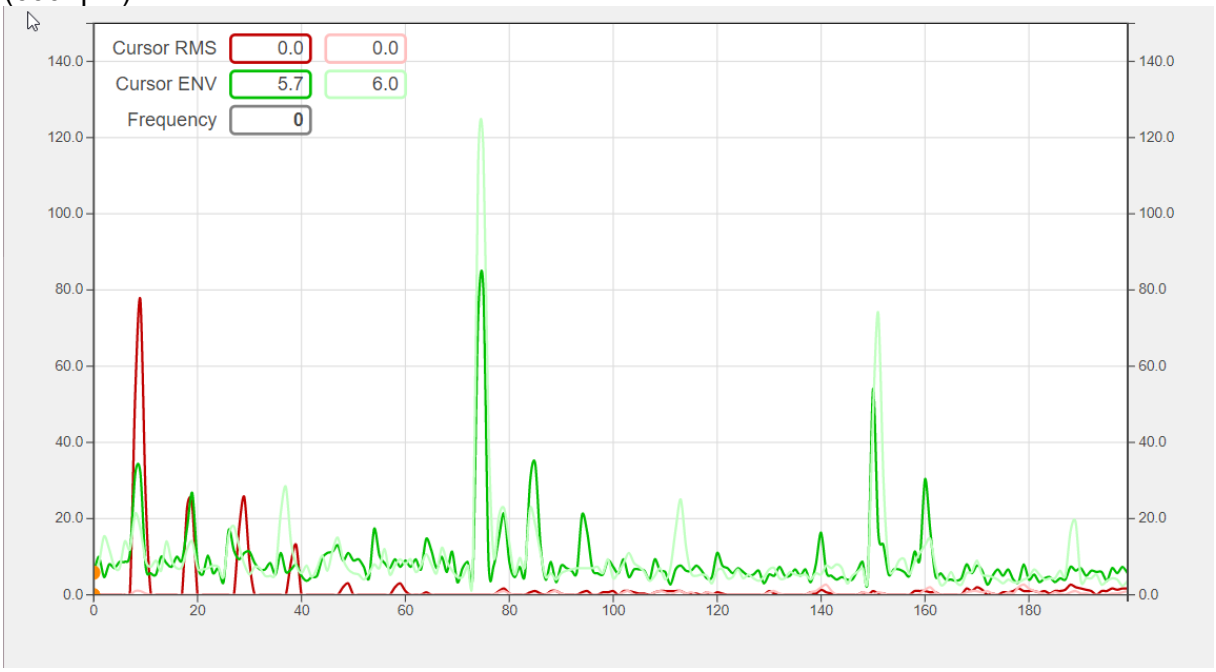
The “Analyse” page shows the core functionality of the library. The visualization shows one module and one channel at a time.

On the top right the module and the channel can be selected as well as the start and stop frequency. Advanced parameters can be changed when necessary.



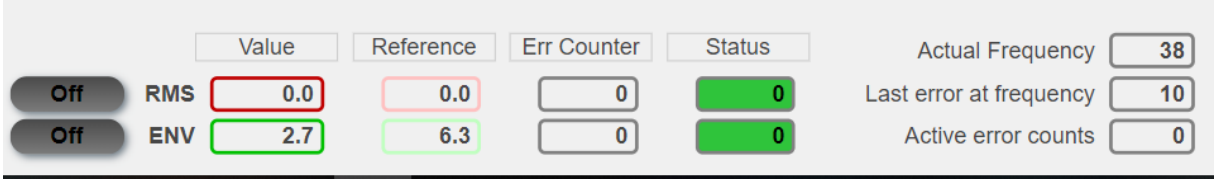
The sampling mode is selected on the lower right corner. Free run is used to just show a graph of the frequency analysis without storing or evaluating the data. Referencing is used to record the machine in the “good state”. Evaluating is the mode where the machine continuously compares the current data against the reference data.

The graph below shows a typical scenario for analyzing the data. The reference data is light green and light red. For this test, an unbalanced load was created and system then ran at 10Hz (600 rpm).

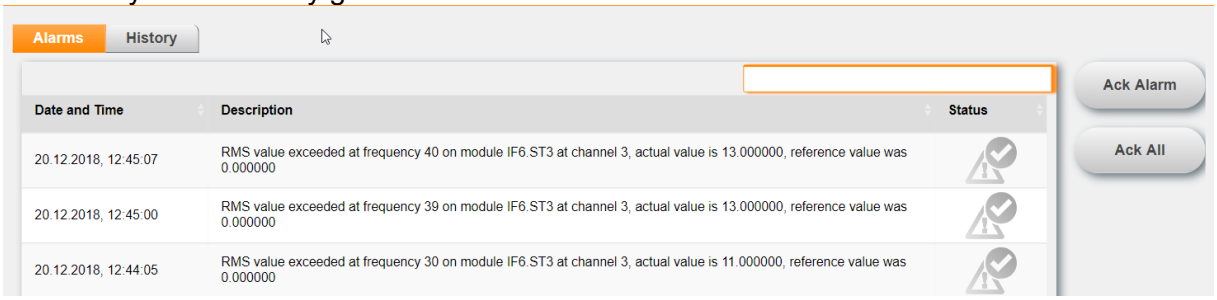


The graph nicely shows how the current RMS (dark red) value exceeds the reference RMS (light red) value at 10Hz and at the harmonics 20Hz, 30Hz...

The bottom shows statistics about the current frequency as well as some additional general information.



The library automatically generates an detailed alarm when one the values exceed the limits.



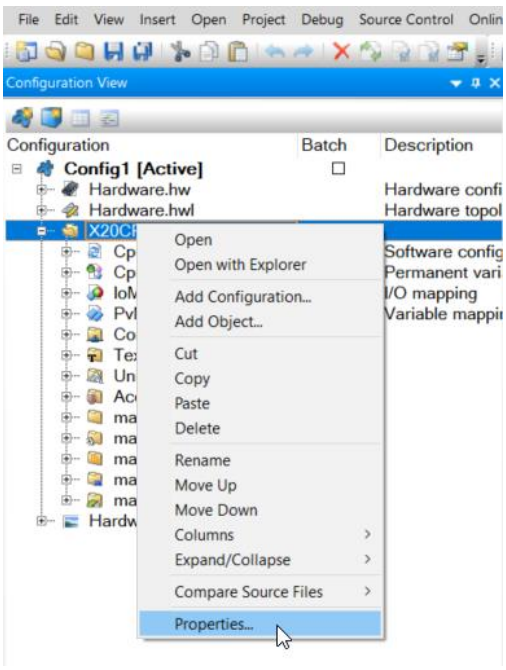
6 Buffer upload

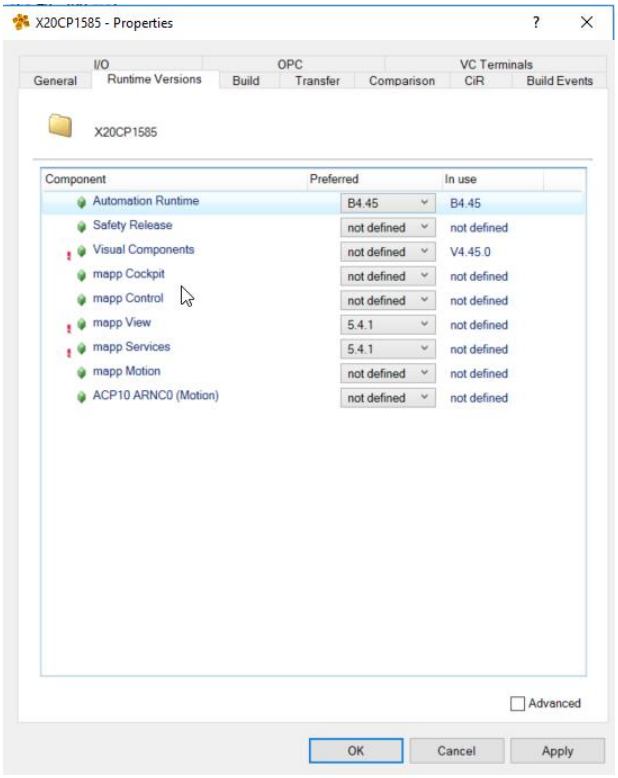
In some cases, it can be required to analyze the raw data of the X20CM4810 module. The task “CM_RawData” provides the necessary sample code to upload and save the raw data from the module. The sample uses the library AsIOVib in the background to retrieve the data from the module. The following structure is used to interface with sample task.

CM_buffer			
	CMD		
		Upload	Start uploading the module buffer
		Cancel	Cancel the uploading process
		ErrorReset	Reset pending errors
	PAR		
		ModulePath	This is the module path in the physical view in Automation Studio
		BufferType	This is the type of buffer used for the raw data, see the Automation Studio help (d1da24ea-d3c8-4381-a835-fd7cc7a95a87) for details.
		SensitivitySensor	Sensor sensitivity for data recording
		FileName	File name for the raw data
		DeviceName	Where the data is stored
		VisuBufferEnable	Used for the visualization to enable/disable controls
		Progress	Percentage of uploaded data
		Status	Status of the uploading library
	DAT		
		xAxis	Time of the datapoint in miliseconds. The total recording time depends on minimum frequency selected in the hardware configuration of the module.
		yAxis	Raw data for the corresponding time
	ERR		
		Text	Error text
		No	Error number
		State	State where the error occured

7 Getting started

7.1 Configure runtime versions





Select mappServices version 5.7.1 or higher.
Select mappView version 5.7.1 or higher when mappView is used as well.

7.2 Import libraries

Import the following B&R libraries into the project.

- AsIOAcc
- AsBrStr
- FileIO
- Standard
- MpRecipe
- MpAlarmX
- MpBase

7.3 Import the project

Import the file CM_Sample.zip into the project for the basic example. Make sure to drag and drop the imported task and library into the PLC configuration.

8 Tips and Hints

8.1 Ignore alarms for certain frequencies

Sometimes is can be necessary to exclude certain frequencies from the evaluation process. To ignore a frequency set the value “AlarmDisabled” in record to true.

4 Appendix

4.1 Revision History

Version 1.20

First public release