APPLICATION EXAMPLE

AC500 CMS PHASE ANALYSIS

SPEED CALCULATION, ORBIT PLOT

Contents

1 Disclaimer

A. For customers domiciled outside Germany /

Für Kunden mit Sitz außerhalb Deutschlands

„Warranty, Liability:

The user shall be solely responsible for the use of this products described within this file. ABB shall be under no warranty whatsoever. ABB's liability in connection with application of the products or examples provided or the files included within this products, irrespective of the legal ground, shall be excluded. The exclusion of liability shall not apply in the case of intention or gross negligence. The present declaration shall be governed by and construed in accordance with the laws of Switzerland under exclusion of its conflict of laws rules and of the Vienna Convention on the International Sale of Goods (CISG)."

„Gewährleistung und Haftung:

Der Nutzer ist allein für die Verwendung des in diesem Dokument beschriebenen Produkte und beschriebenen Anwendungsbeispiele verantwortlich.

ABB unterliegt keiner Gewährleistung. Die Haftung von ABB im Zusammenhang mit diesem Anwendungsbeispiel oder den in dieser Datei enthaltenen Dateien - gleich aus welchem Rechtsgrund - ist ausgeschlossen. Dieser Ausschluss gilt nicht im Falle von Vorsatz oder grober Fahrlässigkeit. Diese Erklärung unterliegt Schweizer Recht unter Ausschluss der Verweisungsnormen und des UN-Kaufrechts (CISG)."

B. Nur für Kunden mit Sitz in Deutschland

„Gewährleistung und Haftung:

Die in diesem Dokument beschriebenen Anwendungsbeispiele oder enthaltenen Dateien beschreiben eine mögliche Anwendung der AC500 bzw. zeigen eine mögliche Einsatzart. Sie stellen nur Beispiele für Programmierungen dar, sind aber keine fertigen Lösungen. Eine Gewähr kann nicht übernommen werden.

Der Nutzer ist für die ordnungsgemäße, insbesondere vollständige und fehlerfreie Programmierung der Steuerungen selbst verantwortlich. Im Falle der teilweisen oder ganzen Übernahme der Programmierbeispiele können gegen ABB keine Ansprüche geltend gemacht werden.

Die Haftung von ABB, gleich aus welchem Rechtsgrund, im Zusammenhang mit den Anwendungsbeispielen oder den in dieser Datei enthaltenen Beschreibung wird ausgeschlossen. Der Haftungsausschluss gilt jedoch nicht in Fällen des Vorsatzes, der groben Fahrlässigkeit, bei Ansprüchen nach dem Produkthaftungsgesetz, im Falle der Verletzung des Lebens, des Körpers oder der Gesundheit oder bei schuldhafter Verletzung einer wesentlichen Vertragspflicht. Im Falle der Verletzung einer wesentlichen Vertragspflicht ist die Haftung jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht zugleich ein anderer der in Satz 2 dieses Unterabsatzes erwähnten Fälle gegeben ist. Eine Änderung der Beweislast zum Nachteil des Nutzers ist hiermit nicht verbunden.

Es gilt materielles deutsches Recht unter Ausschluss des UN-Kaufrechts."

2 Introduction

2.1 Scope of the document

In this example the possibility of phase analysis with the Condition Monitoring System is described. This is mainly used when working with slide bearings. Two analog proximity sensors are mounted with an offset of 90° to a bearing. In addition, also a key phasor is mounted which is detecting one peak per revolution. In the example the phase between both sensor signals is calculated at the rotation speed of the bearing. The signals can be plotted into an orbit plot as well to visualize the movement of the slide bear.

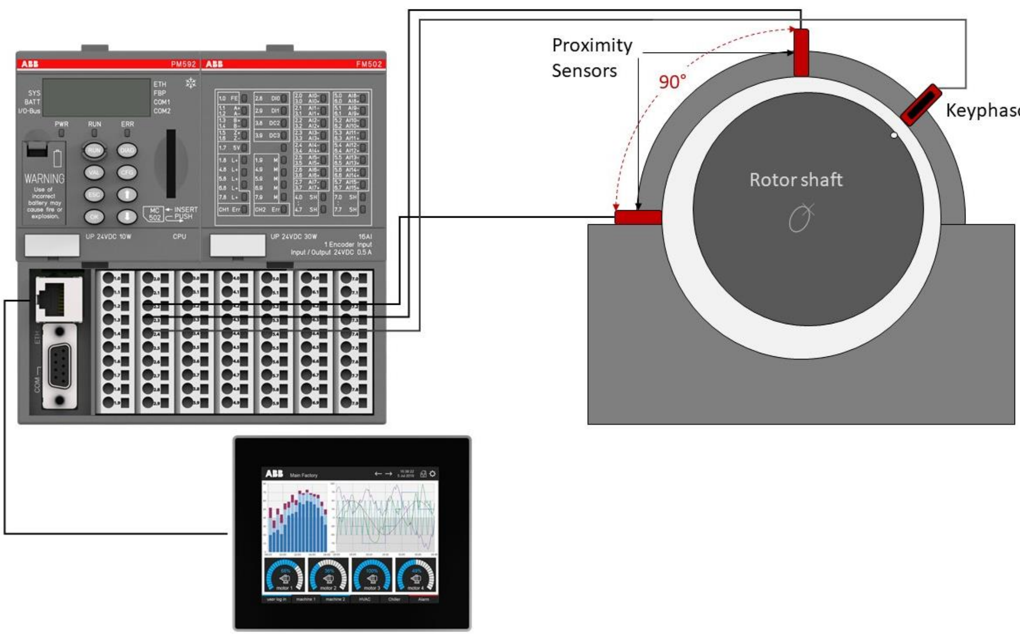
The documentation is defined in four parts. The first three topics deal with the AC500 program. Thereby the speed detection is considered. Afterwards the phase offset calculation is described, as well as the orbit plot calculation in the end. The last chapter describes some settings in the CP600 for the visualization.

2.2 Compatibility

The application example explained in this document have been used with the below engineering system versions. They should also work with other versions, nevertheless some small adaptations may be necessary, for future versions.

- AC500 V2 PM592-ETH and FM502.
- Automation Builder 2.4.1 or newer
- SP Library Version 2.1.0 or newer

2.3 Overview



3 Speed detection

For the phase analysis or orbit plots the exact system speed is necessary. Furthermore, the speed must not variate during the measurement. There are several possibilities how to detect the system speed. In this chapter three different possibilities are explained, and advantages and disadvantages are shown. The table below gives an overview.

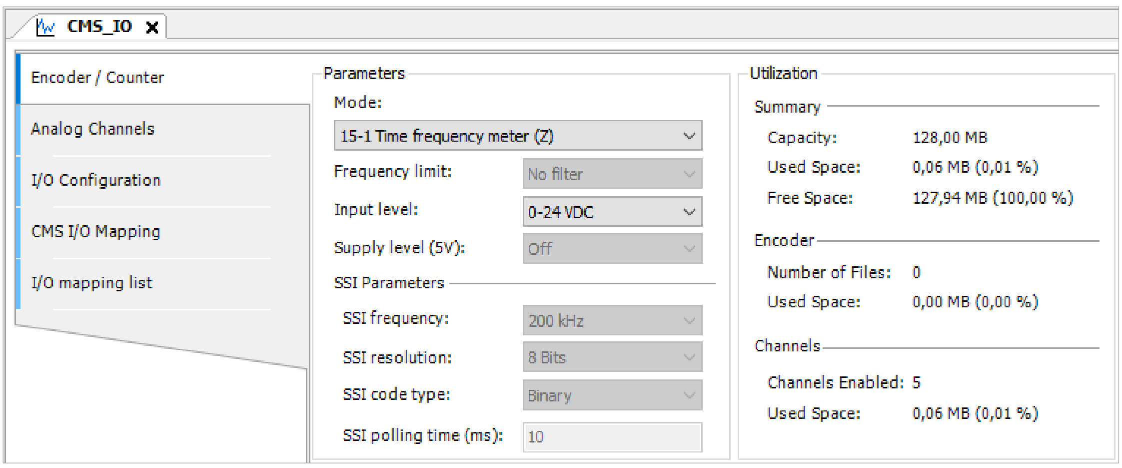
	Drive model	Encoder	Keyphasor
Effort	Low	Medium	High
Accuracy	Low	High	High
Positioning possible	No	No	Yes
Available as wave file	No	No	Yes

3.1 Speed from a drive model

Depending on the used setup the drive can have an output which gives the current speed from a drive model. The speed value of the drive can be read easily without any complex calculation by the drives output. There could be a difference between the calculated speed and the real speed. In addition the calculated speed is not that fast reacting when there is any change on the real drive speed.

3.2 Speed from an Encoder

The FM502 module includes a complete onboard encoder channel. In case the position and direction doesn't matter the Z-channel can be used as time frequency meter. Therefor the Encoder mode 15 is used in the Automation Builder configuration and the function block CMS_IO_FREQ_SCAN is used in the CODESYS Application.



```
encoderFreqScan(
                    EN := TRUE,             (* Always calculate the speed *)
                    INST := ADR(zCMS_IO_INIT0),
                    EN_CNT := TRUE,         (* Counter is enabled *)
                    EN_0 := FALSE,          (* Falling edges are not counted *)
                    EN_1 := TRUE,           (* Rising edges are counted *)
                    EN_FREQ := TRUE);       (* Frequency is calculated *)
```

In the example project is the function block “SpeedEncoder”. With the instance of CMS_IO_FREQ_SCAN the drive speed is calculated. When there hasn’t been a new encoder peak for more than two seconds the timer sets the output to invalid, as the drive might not moving anymore.

With the two triggers it is checked when the measurement is started, and when it is finished. As long as the measurement is ongoing the current speed value is summed up to calculate the average drive speed at the end. Furthermore it is also calculated the minimal and maximal drive speed and the output can be set to invalid in case the speed varied too much.

In contrast to the speed detection from the drive model the encoder is calculating the speed not from a model but from the real system, which makes the value more accurate.

The “SpeedEncoder” function block is inside the program and called but the output is not used.

3.3 Speed from an analog key phasor

It is also possible to use an analog or digital key phasor as analog channel. In case of using a digital key phasor make sure, that its voltage is not higher than 10 V. Otherwise an external voltage divider needs to be used to connect the key phasor to the analog input. The key phasor is detecting any irregularity of the shaft like a hole or screw which will be monitored once per revolution.

The key phasor signal is recorded by the module as wave file and can be analyzed later. The correct speed can be calculated from the peaks in the file. Also small changes in the speed are visible inside.

The first peak of the key phasor and the system speed give the possibility to calculate in which position the shaft was when the recording was started or in case of any disturbance to check in which position the shaft was.

The recorded key phasor has the big advantage, that in addition to the vibration signal also the speed is saved as file. This allows also later to do detailed analysis including the drive speed.

In the example project the function block “SpeedCalculation” is used to analyze the key phasor signal. This block consists of two function blocks. At first the key phasor wave file is read into the local memory. In the next step the data is analyzed to get the average, minimal and maximal speed as well as the time of the first key phasor peak. To calculate this an instance of SP_SPEED_KEYPHASOR_App is used.

4 Phase analysis

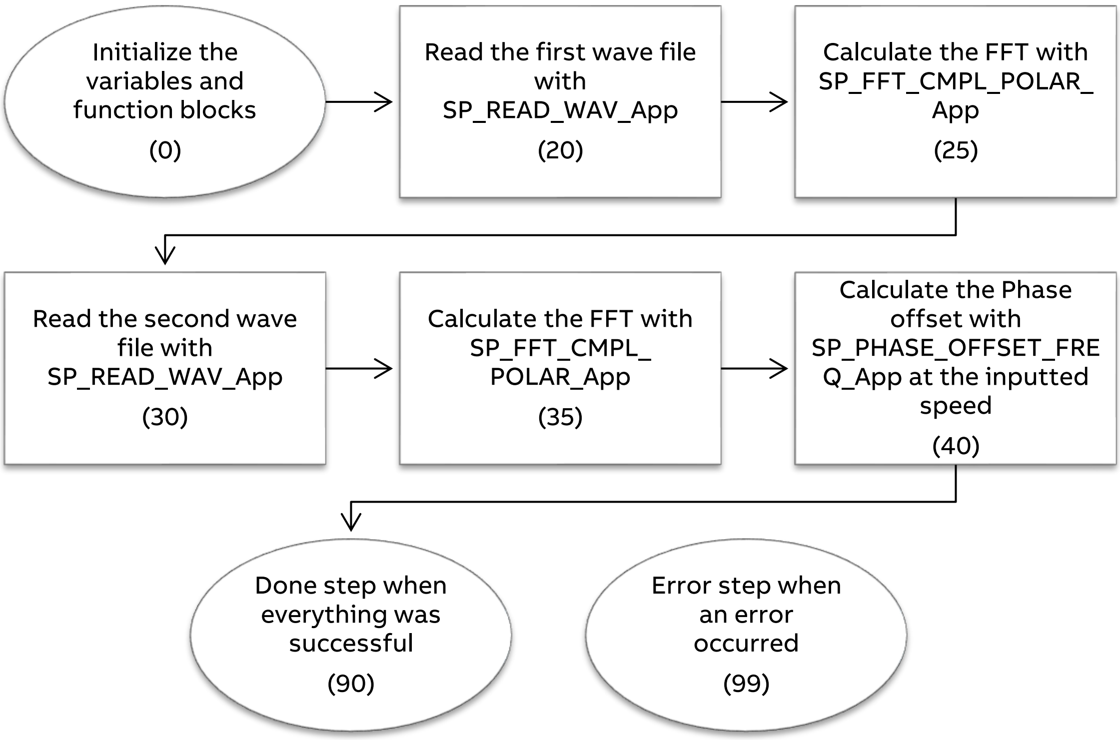
4.1 In the time domain

It is possible to calculate the phase offset between two singals in the time domain. As the signals usually have also some noise they need to be filtered first. A filtering in the time domain is time consuming. Furthermore, the filtering might not output an ideal sinus wave which might make the high point detection difficult. Filtering with the SP_FIR_FILTER_App and phase analysis with the SP_PHASE_OFFSET_TIME_App is possible but not recommended.

4.2 In the frequency domain

In contrast to the phase analysis in the time domain it is also possible to do these calculations in the frequency domain. This has the advantage that no filtering is necessary. To do these calculations the FFT calculation must give the phase information in addition to the magnitudes. To find the right position the speed must be very exact.

In the program the function block “PhaseAnalysis” is used. The program itself is performed in a step chain.



Whenever the exact speed is known it is recommended to use the analysis in the frequency domain instead of the time domain analysis. The frequency domain calculations take less time and are more accurate.

As the sensors are mounted with 90° also the phase offset between the two channels should be around 90°. In case the drive is moving in the other direction the phase offset should be about -90°.

5 Orbit calculation

In addition to the phase information it could be also important to see how round the shaft is sliding in the oil film. The position of the shaft can be plotted into an orbit. Therefore, the first channel is used as X-axis and the second channel as Y-axis. Depending on the signals and the noise it could also be possible that the orbit is too noisy to see anything. Therefore, a FIR filtering is done at first. Depending on the length of the signal, this filtering might be very time consuming.

In the function block "OrbitCalc" these calculations are done in a step chain.

In the steps 10 to 21 both wave files are read into the local memory and afterwards FIR filtered. During the read the signals are scaled a bit bigger as the CP600 scatter diagrams cannot plot values below one correctly. Furthermore the CP600 scatter diagram can only work with arrays up to 1000 samples. That's why only the first 1000 samples are taken into account.

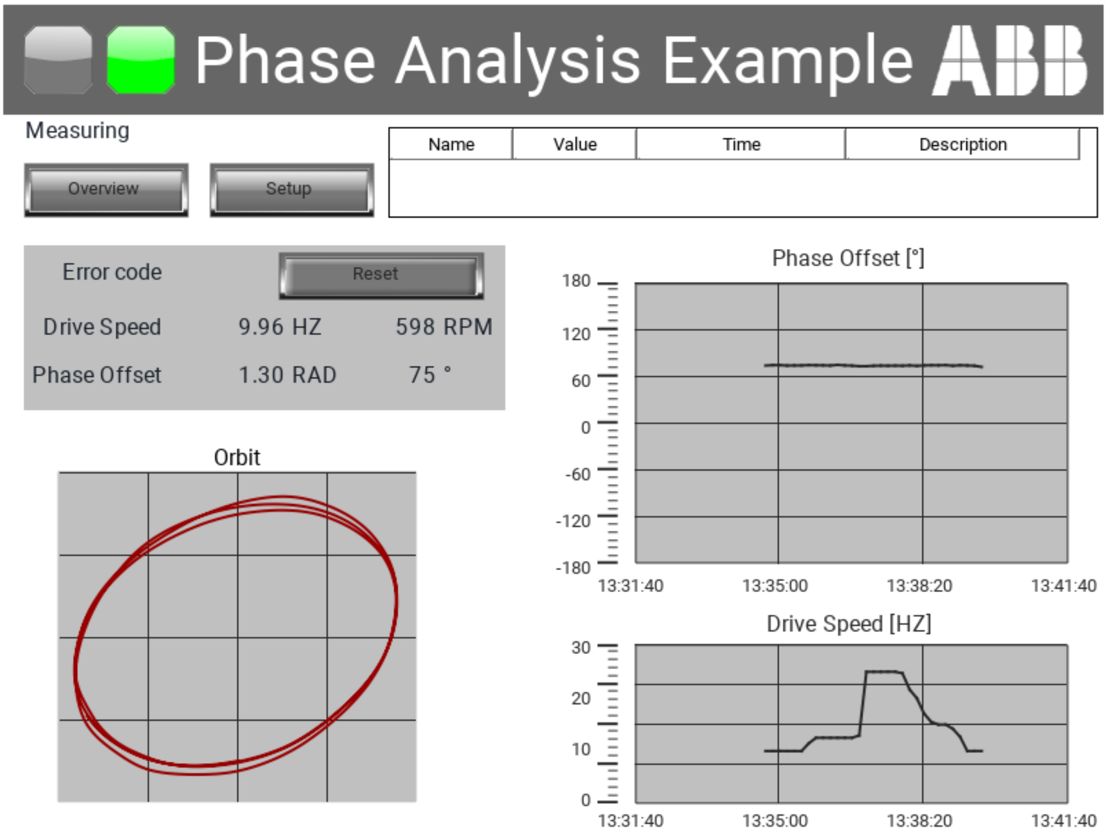
In step 30 of the step chain it is furthermore checked if the signal has at least 1500 samples. If this is the case not the first 1000 samples but the samples from 500 to 1500 are taken into account. This has the reason that due to the FIR filtering the first 256 values are smaller. The arrays where the X and Y information should be stored is an input of this function block. The selected data is copied in the arrays.

> In case of using high sampling rates the first 1500 values might too less datapoints to draw several rotations. Therefore, use the resampling functionality to down sample the original file to a file containing less datapoints. This way the whole measurement can be shown but with less accuracy.

In addition, also the absolute maximal value is calculated to use this as max value for the plot in the CP600.

6 CP600 visualization

For the visualization a CP6410 panel is used. But the program can also be used for all other CP600 panels.



6.1 Tags in CP600

For the communication between AC500 and CP600 the ABB CoDeSys ETH protocol is used. In addition, also Modbus TCP is used, as arrays for the scatter diagram cannot be a tag in the CODESYS ETH protocol.

Also some internal variables are used. These are explained later.

PLC	Configuration	Dictionaries
ABB CoDeSys ETH:prot1	CfgVer=1 prefix=PLC_AC500_V2 ip_Address=192.168.6.10 Port=1200 Block	[ABB CoDeSys ETH prot1] Application
Modbus TCP:prot2	CfgVer=1 ipAddress=192.168.6.10 port=502 UDP=false ENCAP=false timeou	None available...
Variables:prot3	CfgVer=1	None available...

6.2 KPI values

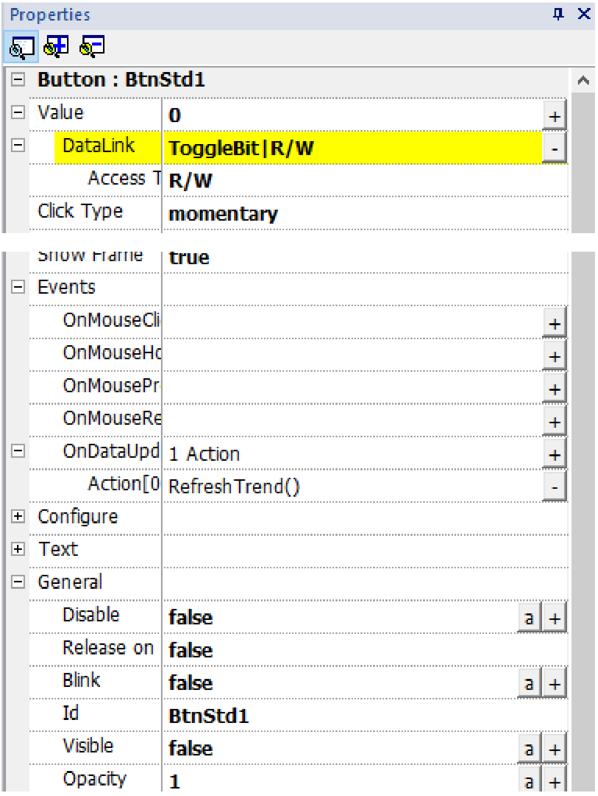
The main page “Overview” is shown as default after starting the panel. Below the status bar in the upper left corner, the KPI values are shown. These are the Error code, the drive speed in Hz and RPM as well as the phase offset in radians and degree.

Depending on the application it could be also useful to display RMS or further KPI values here.

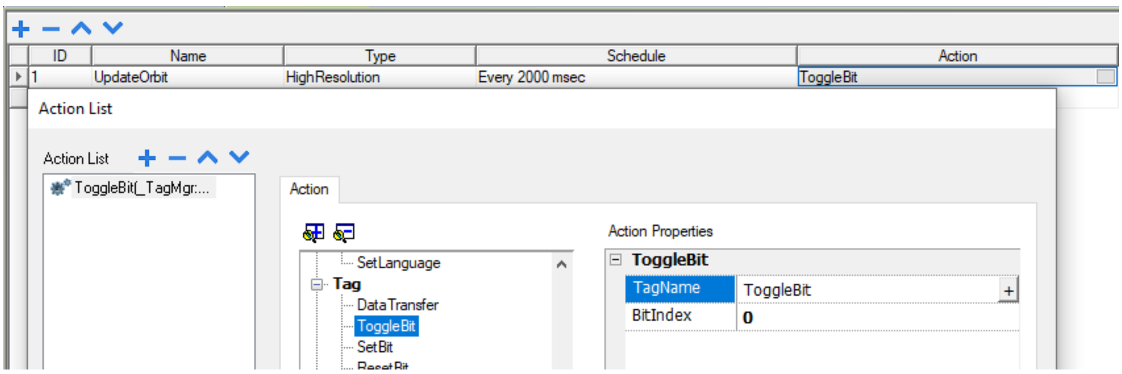
6.3 Orbit plot

Below the KPI Value on the lower left corner the orbit is plotted. Therefore, a scatter diagram is used. The variables xAxis, yAxis as well as lrMaxPlot are connected to the graph.

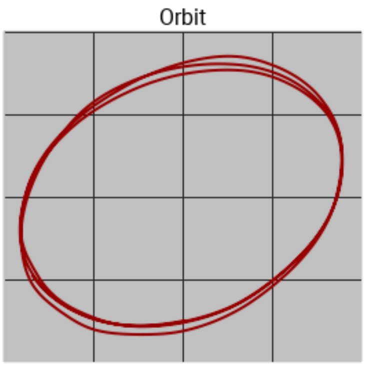
The scatter diagram itself is only updating when it is loaded new to do an update automatically a scheduler and an invisible button are used. The button which is visible in the lower left corner is linked to the intern ToggleBit. In the General setting the visibility is false. As Event an OnDataUpdate action is added. In this action RefreshTrend is chosen and the ScatterDiagram is seleted.



In the next step this invisible button is triggered automatically. Therefore, the scheduler is used. The first scheduler named UpdateOrbit has a scheduling time of two seconds. This time can also be adapted for your application needs. As action the internal ToggleBit is toggled.

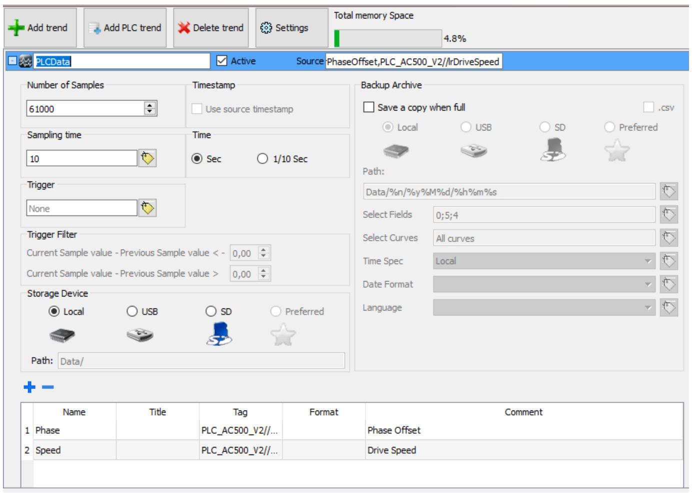


This way the orbit is updated automatically.



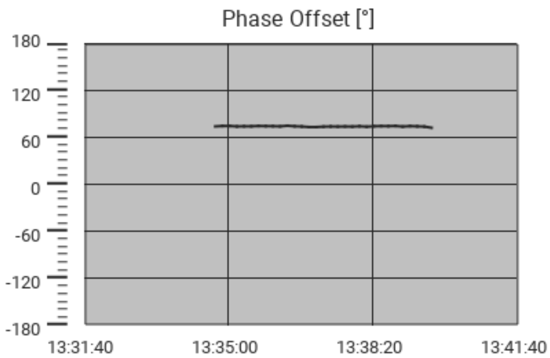
6.4 Trends

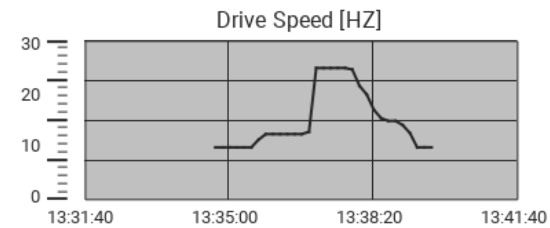
The drive speed and phase offset are shown as KPI values but also historical values of these KPIs should be visible. Therefore, a trend is used.



The trend PLCData is sampling the phase and the speed each 10 seconds.

The two trends are also shown on the main screen. The X-Axis duration is set as 10 minutes but can also be changed.



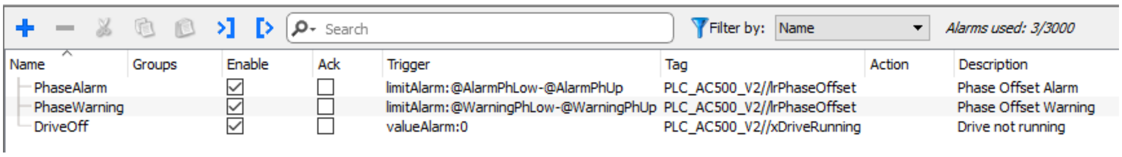


6.5 Alarms

An alarm window is shown in the status bar on the top.

In the alarms three alarms are added.

The first is the alarm, that the phase offset is too high or too low. The second is the warning that the phase offset is high or low. The third is a notification that the drive is not running.

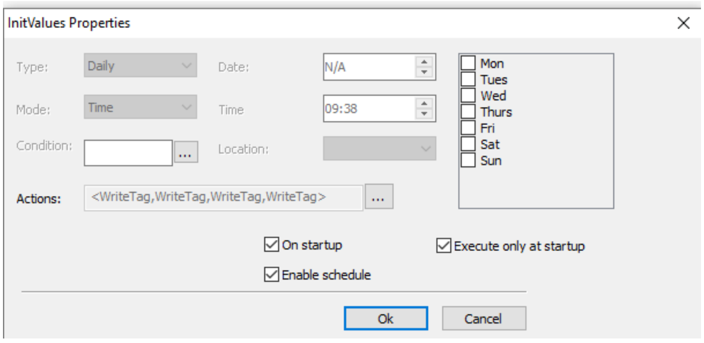


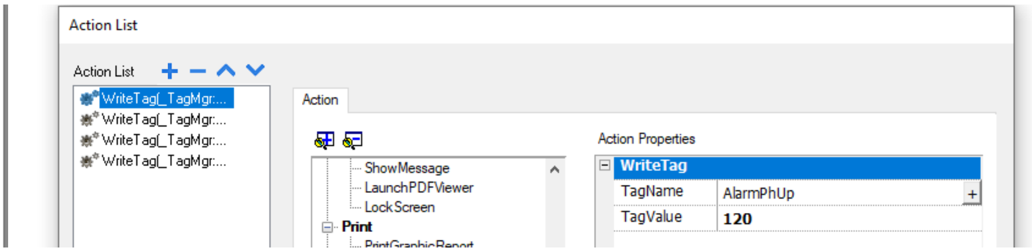
The alarms regarding the phase are checked with upper and lower limits. The running drive is just a binary check. Depending on the alarms different colors were used.

The phase checks should only be done in case the drive is running. Otherwise, the value 0 will be always an alarm. To archive this the xDriveRunning variable is used as Remote Enable.

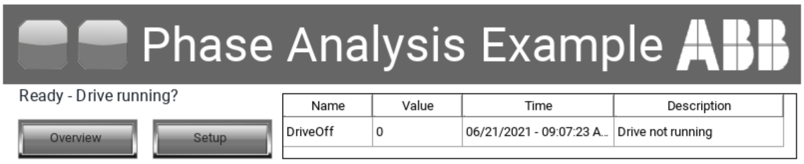
Property	Value
Name	PhaseAlarm
Groups	
Enable	false
Ack	false
Reset	false
Buffer	AlarmBuffer1
Trigger	limitAlarm:@AlarmPhLow-@AlarmPhUp
Tag	PLC_AC500_V2//lrPhaseOffset
Remote Enable	PLC_AC500_V2//xDriveRunning
Remote Ack	none
Ack Notify	none
Action	
User Action	
Description	Phase Offset Alarm
Color	...
AckBlink	false
Severity	2-below normal
Events	76,76,1,1
Custom Field 1	
Custom Field 2	

On the page “Setup” the limits for the phase offset warnings and alarms can be changed during runtime. The initialization of these tags is done with a scheduler, that is only executed during HMI start.





During runtime when the drive is not running the according info is shown.



—

ABB Automation Products GmbH
Eppelheimer Straße 82
69123 Heidelberg, Germany
Phone: +49 62 21 701 1444
Fax: +49 62 21 701 1382
E-Mail: plc.support@de.abb.com
www.abb.com/plc