Calculating production figures using WinCC standard functions

STEP 7 / WinCC

Application Description • December 2009

Applications & Tools

Answers for industry.

SIEMENS

Industry Automation and Drives Technologies Service & Support Portal

This entry is taken from the Internet Service Portal of Siemens AG, Industry Automation and Drives Technologies. Clicking the link below directly displays the download page of this document.

http://support.automation.siemens.com/WW/view/en/38701615

For questions about this document, please use the following e-mail address:

online-support.automation@siemens.com

SIEMENS

SIMATIC WinCC
WinCC KPI Analyze

Warranty and Liability

Note

The application examples are not binding and do not claim to be complete regarding configuration, equipment and any eventuality. The application examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These application examples do not relieve you of the responsibility to use sound practices in application, installation, operation and maintenance. When using these application examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these application examples at any time without prior notice. If there are any deviations between the recommendations provided in this application example and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). However, claims arising from a breach of a condition which goes to the root of the contract shall be limited to the foreseeable damage which is intrinsic to the contract, unless caused by intent or gross negligence or based on mandatory liability for injury of life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Table of Contents

1 Automation Task

1.1 Overview

Introduction

This sample application, shows how the plan holding time, utilization time and the breakdown times of a machine are detected with WinCC. The KPI value (Key Performance Indicator) of the machine is then calculated from these specified values. The production figure is to be considered as a tool to increase productivity.

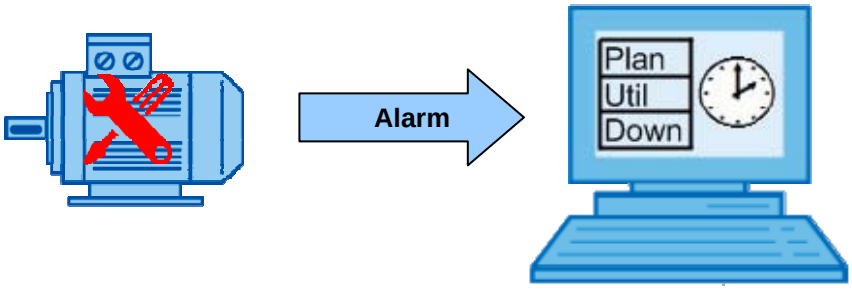
The following functions are used in the sample project:

- block generated in SCL (simulation of a motor)
- evaluation of WinCC alarm message system
- access to database via VB script

Overview of the automation task

The figure below shows an overview of the automation task.

Figure 1-1



Description of the automation task

The example describes the calculation of the KPI, using a simulated machine. The plan holding, utilization and breakdown times are detected via the WinCC alarm message system. The simulation of the machine is via a block in the S7 program which triggers the respective alarm messages.

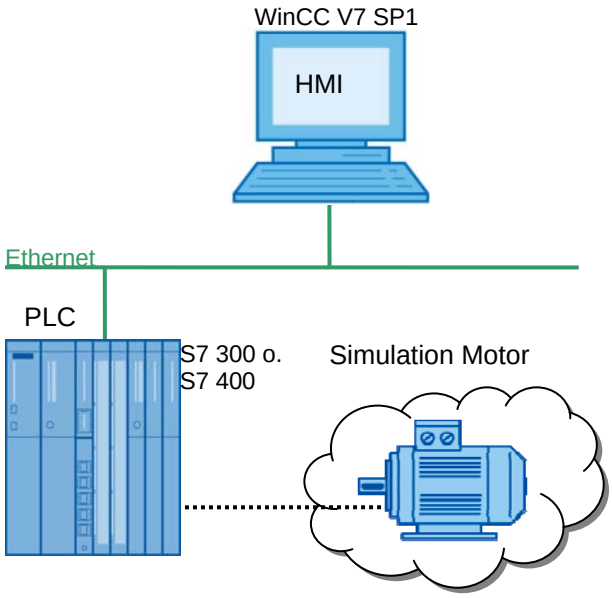
2 Automation Solution

2.1 Overview of overall solution

Schematic layout

The following figure displays the most important components of the solution:

Figure 2-2



Topics not covered by this application

This application does not contain a description of

- STEP 7
- WinCC
- Visual Basic
- Microsoft SQL

Basic knowledge of these topics is assumed.

2.2 Description of the core functionality

A motor is simulated in the S7 program which sends the alarm messages "Machine Power On", "Machine Running" and "Motor Failed" based on the events. Operation is via WinCC.

The alarm messages are displayed and evaluted in WinCC Alarm Control (hit list).

Calculating the KPI is performed in WinCC by VB script. The script accesses the database table temporarily generated by WinCC Alarm Control. The following data is detected:

- plan holding time
- utilization time
- breakdown time
- frequency of breakdowns

From the information of the alarm archive the following values are calculated:

- Utilization
- Availability
- MTBF (Mean Time Between Failure)
- MTTR (Mean Time To Recover)

Advantages of this solution

The solution introduced here offers you the following advantages:

- detection of production figures without additional software
- no additional licences are necessary

2.3 Hardware and software components used

The application was generated with the following components:

Standard software components

Table 2-1

Components	No.	MLFB / order number	Note
SIMATIC S7 V5.4	1	6ES7810-4CC08-0YA5	
SIMATIC S7 SCL	1	6ES7811-1CC05-0YA5	(Optional) for programming of blocks in SCL
SIMATIC CFC V7.0	1	6ES7658-1EX07-2YA5	(Optional) generating the S7 program
SIMATIC WinCC V7.0 SP1 RC 128	1	6AV6381-2BM07-0AX0	
SIMATIC S7-PLCSIM V5.4	1	6ES7841-0CC05-0YA5	(Optional) To simulate process control systems

Sample files and projects

The following list contains all files and projects used in this example.

Table 2-2

Components	Note
WinCC_KPI_Analyze_V10.zip	This zip file contains the STEP 7 project and the integrated WinCC project
38701615_WinCC_KPI_Analyze_d.pdf	This document.

Alternative solutions

For SIMATIC WinCC the options "SIMATIC WinCC/DowntimeMonitor V7.0" and "SIMATIC WinCC/ProcessMonitor V7.0" are available to detect, display and evaluate production figures. This evaluation allows an increase of plant productivity.

Both options are componets of SIMATIC WinCC/Plant Intelligence. Pre-defined key performance indicators (KPI) can be detected and visualized by individual machine modules, units and production lines (equipment). The most important properties are:

- tracing for breakdown times
- assignment of causes and reasons
- evaluation and monitoring of equipment efficiency.
- decision-making on the basis of performance indicators by identifying the events that cause cost-intensive equipment failures.

http://support.automation.siemens.com/WW/view/en/34519922

The "SIMATIC Maintenance Station 2009" option for the SIMATIC WinCC process visualization system offers the possibility to diagnose and maintain machines and plants in a central system to support preventive, status-based maintenance and reactive maintenance meassures.

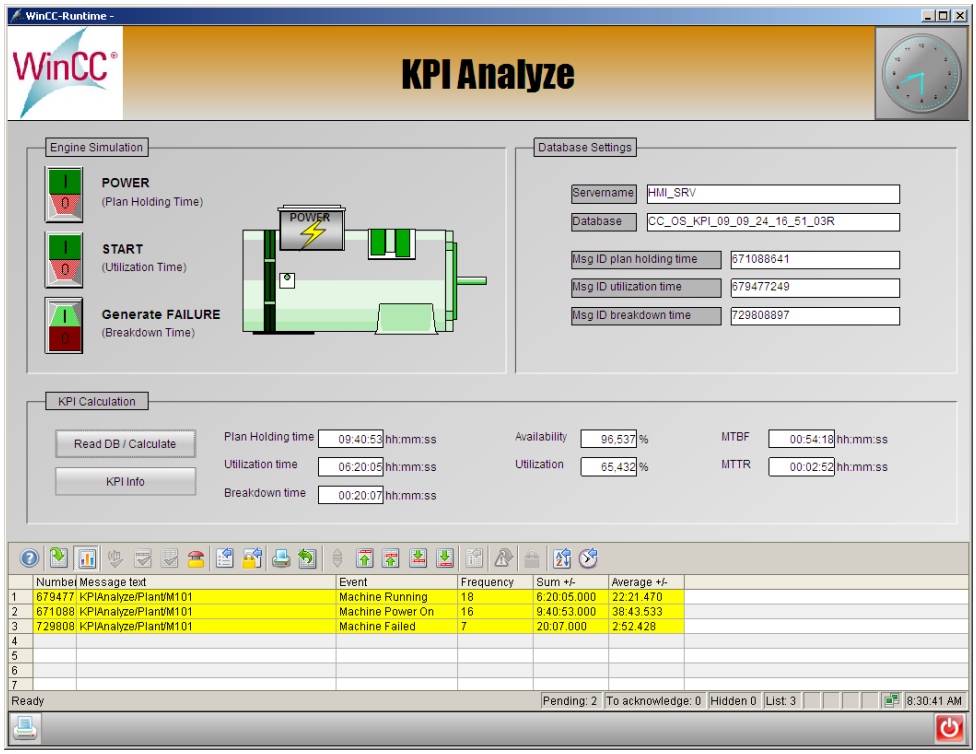
http://support.automation.siemens.com/WW/view/en/31238198

3 Function Mechanisms of this Application

3.1 General overview

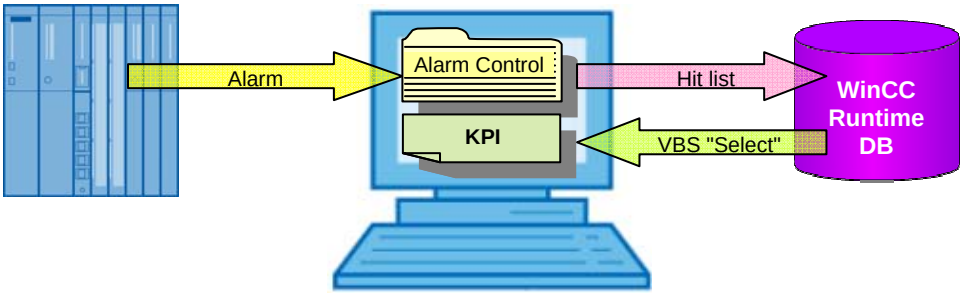
The figure shows the simulated motor with the operating options in the "Engine Simulation" area. The messages generated by this simulated motor are displayed in the alarm control in the lower area of the screen. The "Database Settings" area displays the necessary parameters for database inquiries. After clicking the "Read DB / Calculate" button in the "KPI Calculation" area, the necessary data is read from the database and displayed in the process display.

Figure 3-3



3.2 Functionality

Figure 3-4



If an alarm is triggered in the S7 program, there will be an entry in the alarm logging of the WinCC Runtime database. In the sample project this is done via the buttons of the simulated motor.

The hit list of the WinCC alarm control displays the duration and frequency of alarms. These values are the basis for caculating the production figures.

The values of the hit list are stored in the Runtime database in a temporary database table "AlgHitListResult_tmp". Since this database table is available in uncompressed form, it is sufficient to use the Microsoft "SQLOLEDB" provider for database inquiries. WinCC archive data is usually stored in compressed form in the database. This compressed data can only be read out with the optional "WinCCOLEDB" provider. The "WinCCOLEDB" provider is a component of "WinCC/Connectivity Pack". Information regarding the "WinCC/Connectivity Pack" can be found under the following address.

http://support.automation.siemens.com/WW/view/en/28887620

Once the necessary values were read out from the database, the production figures (KPI) are calculated. Reading out from the database and the calculation of the KPI is programmed in the same VB script. The script is executed by pressing the "Read DB / Calculate" button.

4 Configuration Process

Note The sample project was created with the help of the "S7-PLCSIM" S7 simulator using a S7-400 controller. If you would like to use the project in a real environment, please adjust your hardware configuration to your conditions.

4.1 Simulation block "Machine"

The block was programmed in SCL. The block is only supposed to simulated the feedback messages of a motor such as "Power", "Run" and "Failure" and to transfer them to WinCC, using the cronological reporting procedure.

The following functions were implemented:

- operation via WinCC
- trigger of meassages (Alarm_8P)

Note When operating a S7-300 CPU, the use of a chronological message procedure is not possible. In this case it is necessary to use the bit message procedure instead.

4.1.1 Tag declaration

The following input tags were created which are transferred to WinCC using the OS compilation function. With these tags the operation is performed in WinCC.

Table 4-3

Input	Type	Description
POWER	BOOL	Commissions the plant
RUN	BOOL	Starts the motor
FAILURE	BOOL	Simulates motor failure

The following output tags were created which are transferred to WinCC using the OS compilation function. These tags trigger the alarm messages and display the status in WinCC.

Table 4-4

Output	Type	Description
QPow	BOOL	Plant in operation (plan holding time)
Qrun	BOOL	Motor running (utilization time)
Qerr	BOOL	Motor failure (breakdown time)

4.1.2 Program logic

The following programming sets the status of the motor, using the selected settings in WinCC and triggers the respective messages:

Figure 4-5

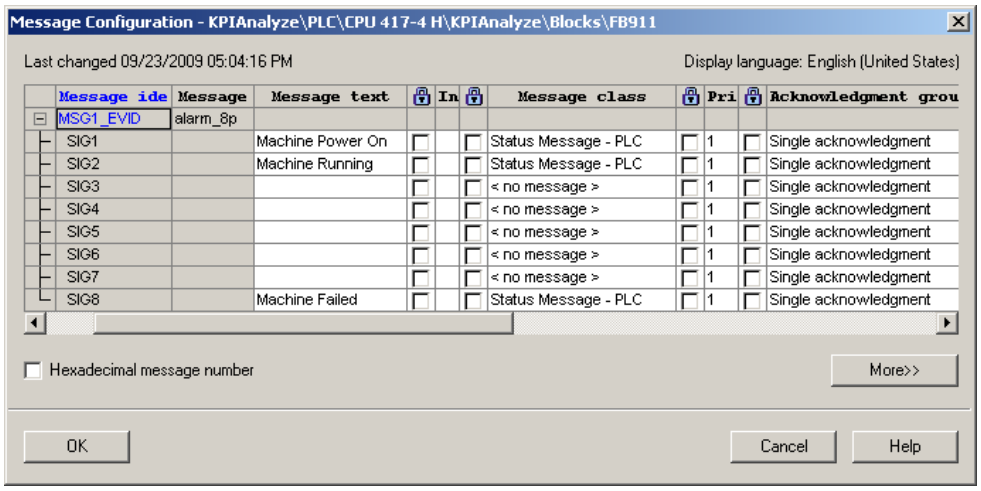
```
BEGIN;
// Turn machine in "ON" state
     QPow := POWER;

// Turn machine in "ERROR" state
     QErr := QPow AND FAILURE;

// Turn machine in "RUN" state
     QRun := QPow AND RUN AND NOT QErr;

     A8P(
          EN_R := 1,
          SIG_1 :=QPow, // Message: Power On
          SIG_2 :=QRun, // Message: Motor runs
          SIG_3 :=0,
          SIG_4 :=0,
          SIG_5 :=0,
          SIG_6 :=0,
          SIG_7 :=0,
          SIG_8 :=QErr, // Message: Motor Failed
          ID := w#16#eeee,
          EV_ID := MSG1_EVID,
          SEVERITY := w#16#40
     );
     MSG1_bDone := A8P.DONE;
     MSG1_bError := A8P.ERROR;
     MSG1_wState := A8P.STATUS;
     MSG1_wAck := A8P.ACK_STATE;
END_FUNCTION_BLOCK
```

Note To be able to use the block in a program of a S7-300 CPU remove the declaration and the call of the "Alarm8P" block and recompile the source code. To trigger the messages in WinCC the bit tags "Qpow", "Qerr" and "Qrun" can be used.

4.1.3 Configuration of message

The configuration of messages is performed on the compiled block in the "Message Configuration" dialog.

Figure 4-6



The texts in the "Message Text" column are generated in WinCC alarm logging as "User Text block – 3". The message class "Status Message – PLC" displayed on the screen corresponds to message class 16 type 253 in WinCC. To be able to select the "Status Message – PLC" message class, the option "Single acknowledgment" has to be deselected.

The correlation between S7 and WinCC message classes is explained in this entry:

http://support.automation.siemens.com/WW/view/en/31622970

Note

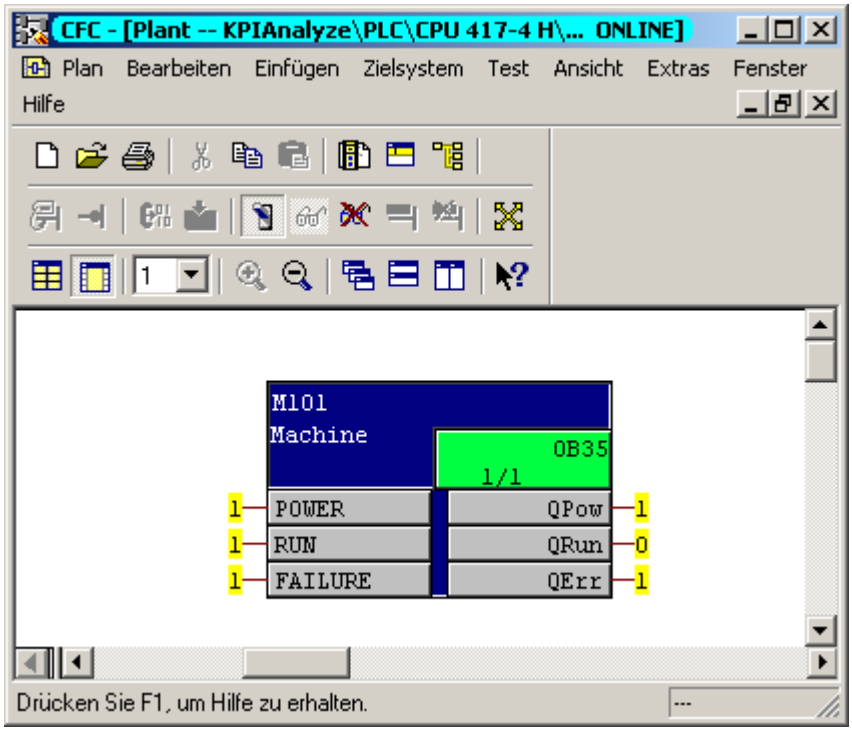
When using the bit message procedure the tags "Qpow", "Qrun" and "Qerr" can be used to trigger a message.

The messages are configured manually in the WinCC alarm logging dialog.

4.2 S7 program

The S7 program was created using the STEP 7 Option CFC. To do this, the compiled "Machine" block was added in the CFC plan "Plant" and the program was compiled.

Figure 4-7



Other configurations are not necessary for this example.

4.3 WinCC project

Since the WinCC project is integrated in STEP 7, the project and all necessary tags, messages (not bit message procedures) and connections were created by the "Compile" function of the WinCC application in the SIMATIC manager.

4.3.1 Configuration of message

To correctly display the messages created when OS compiling in WinCC, the WinCC Alarm Logging was configured as follows:

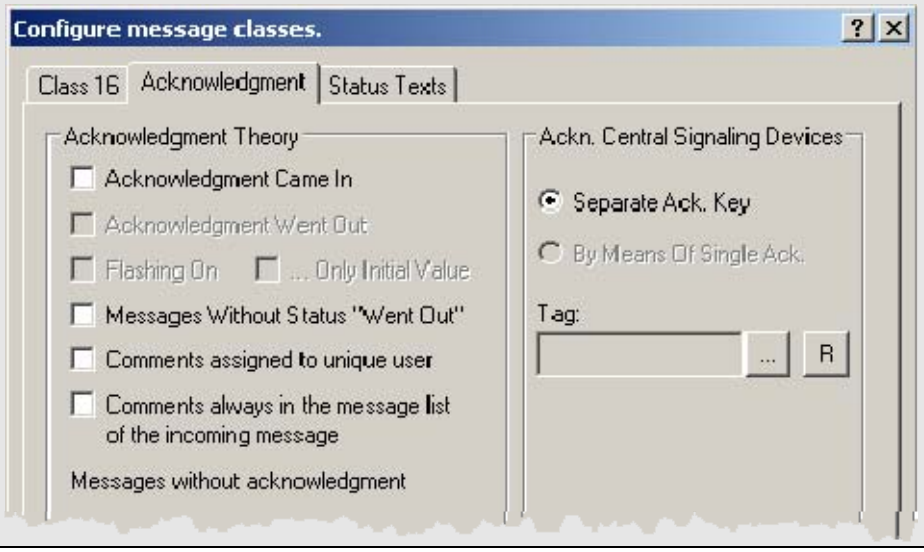
- Usertextblock – 3
  - Name: "Event"
  - Length: 30 characters
- Message class 16
  - Name: "PLC State"
  - Message type 253 (Name: "Engine")

Notes

- These settings are not necessary when you create the WinCC project using the OS project editor. In this case they are automatically made.
- If you are using the bit message procedure, the messages have to be additionally configured in the WinCC Alarm Logging.

CAUTION

To be able to correctly record the time for calculating the production figures it is important that the options "**Acknowledgment Came in"** and **"Messages Without Status "Went Out""** are **not** enabled in the "Acknowledgment" tab.



4.3.2 WinCC Pictures

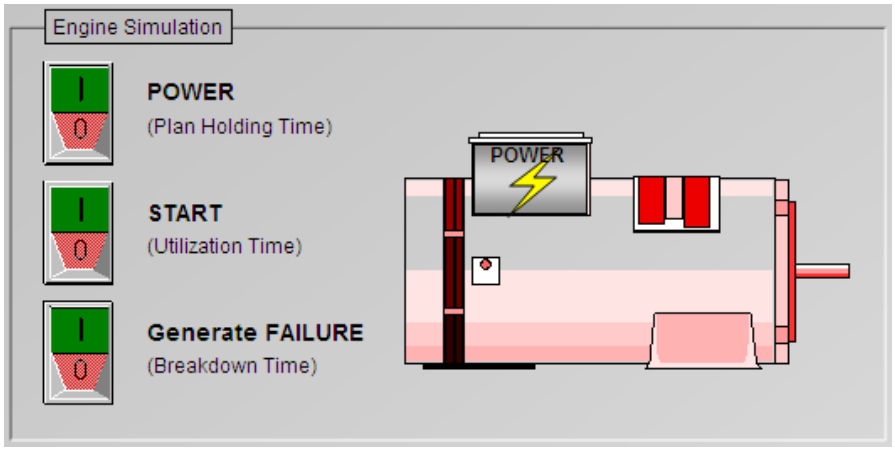
In the motor simulation area, three buttons are configured which have the following functions:

Table 4-5

Switches	Process tag	Instruction
POWER	“POWER“	• The message "Machine Power On" is pending • Plan holding time is recorded
START	“RUN“	• The message "Machine Running" is pending • The utilization time is recorded
Generate FAILURE	“FAILURE“	• The message "Machine Running" is not pending • The message "Machine Failed" is pending • Breakdown time is recorded

The status of the motor is displayed by the motor icon.

Figure 4-8



In the area of the data base settings the necessary settings for a database search are displayed or requested.

Figure 4-9

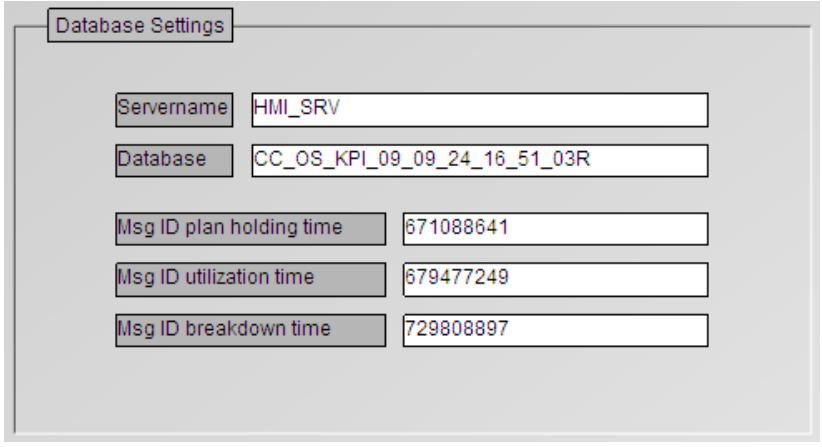


Table 4-6

Name	Process tag	Description
Server name	« @Servername“	Displays the server name at which WinCC Runtime was started.
Database	"@DatasourceNameRT"	Displays the database name of WinCC Runtime.
Msg ID plan holding time	"MsgIDPlan"	Requires the ID input of the message which records the plan holding time. Event: "Machine Power On"
Msg ID untilization time	"MsgIDUtil"	Requires the ID input of the message which records the utilization time. Event: "Machine Running"
Msg ID breakdown time	"MsgIDErr"	Requires the ID input of the message which records the breakdown time. Event: "Machine Failed"

In the area for calculating the production figures the read out data from the database and the calculated values are displayed. Read out and calculation is via the "Read DB / Calculate" button. With help of "KPI Info", a screen can be displayed which shows the formulas used for the KPI – calculation.

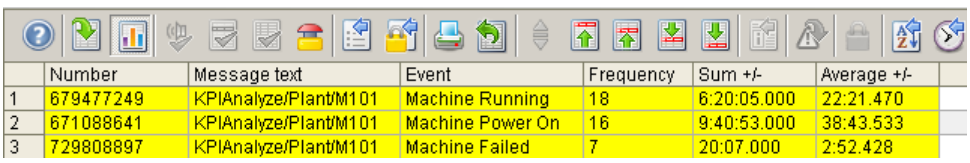
Figure 4-10

Table 4-7

Name	Process tag	Description
Plan holding time	“T_Plan“	Displays the plan holding time.
Utilization time	“T_Util“	Displays the utilization time.
Breakdown time	“T_DecDis“	Displays the breakdown time.
Availability	“Availability“	Displays the availability in %.
Utilization	“Utilization“	Displays the utilization in %.
MTBF	“MTBF“	Displays the average time between failures
MTTR	”MTTR“	Displays the average time for recovery

In the lower area of the display is the configuration of a WinCC alarm control to display the messages and the hit list. This is where the message IDs for database inquiries can be read out.

Figure 4-11



	Number	Message text	Event	Frequency	Sum +/-	Average +/-
1	679477249	KPIAnalyze/Plant/M101	Machine Running	18	6:20:05.000	22:21.470
2	671088641	KPIAnalyze/Plant/M101	Machine Power On	16	9:40:53.000	38:43.533
3	729808897	KPIAnalyze/Plant/M101	Machine Failed	7	20:07.000	2:52.428

4.4 VB script for KPI calculation

On the "Read DB / Calculate" button a VB script is configured which fulfills the following tasks:

- database inquiry by "SQLOLEDB" – Provider
- calculation of KPI based on the detected values
- conversion of seconds values in readable time details

4.4.1 Datenbase inquiry

Table 4-8

No.	Instruction

1.	The screens shows the tag declaration for the database connection and the request string.
	
	`'*************************************************************************`
	`'**** Tags for DB connection                                                ****`
	`'*************************************************************************`
	`Dim strProvider          'Provider for DB connection`
	`Dim strSecure            'Security setting`
	`Dim strDatabase          'WinCC Runtime Database`
	`Dim strServer            'WinCC Runtime Server`
	`Dim strConnectionString 'Connection string`
	`Dim objConnection`
	`Dim objCommand`
	`Dim objRecordset`
	
	`'*************************************************************************`
	`'**** Tags for database query                                               ****`
	`'*************************************************************************`
	`Dim lngMsgIDPlan     'Message ID plan holding time`
	`Dim lngMsgIDUtil     'Message ID utilization time`
	`Dim lngMsgIDErr      'Message ID breakdown time`
	`Dim strSQL           'Query string`
	`Dim lngCount         'Count of columns`

2.	After the declaration of the internal VBS tags the string for the database connection is composed and the message IDs from the WinCC tags are read.
	If the input fields for the message IDs are not filled in in the WinCC screen and if they are preassigned with ‘0’ by default, then a message is emitted and the function is exited.
	
	`'*************************************************************************`
	`'**** Create the connection string                                          ****`
	`'*************************************************************************`
	`strProvider="Provider=SQLOLEDB.1; "`
	`strSecure  ="Integrated Security=SSPI; Persist Security Info=false; "`
	`strDatabase="Initial Catalog=" & HMIRuntime.Tags("@DatasourceNameRT").Read & "; "`
	`strServer  ="Data Source=" & HMIRuntime.Tags("@ServerName").Read & "\WinCC"`
	`strConnectionString = strProvider & strSecure & strDatabase & strServer`
	
	`'*************************************************************************`
	`'**** Read MSG-ID's from HMI input                                          ****`
	`'*************************************************************************`
	`lngMsgIDPlan =HMIRuntime.Tags("MsgIDPlan").Read 'Message ID for plan holding time`
	`lngMsgIDUtil =HMIRuntime.Tags("MsgIDUtil").Read 'Message ID for utilization time`
	`lngMsgIDErr  =HMIRuntime.Tags("MsgIDErr").Read  'Message ID for breakdown time`
	
	`If lngMsgIDPlan = 0 Or lngMsgIDUtil=0 Or lngMsgIDUtil=0 Then`
	`    MsgBox "Missing of Message ID"`
	`    Exit Sub`
	`End If`

3.	The next step is to create a connection to the database.
	`'*****************************************************************************`
	`'**** Create the objects for DB connection                                  ****`
	`'*****************************************************************************`
	`Set objConnection = CreateObject("ADODB.Connection")`
	`Set objRecordset = CreateObject("ADODB.Recordset")`
	`Set objCommand = CreateObject("ADODB.Command")`
	
	`'*****************************************************************************`
	`'**** Open the DB connection                                                  ****`
	`'*****************************************************************************`
	`objConnection.open strConnectionString`
	`objCommand.ActiveConnection = objConnection`

4.	The code section below shows the database inquiry by message ID of the plan holding time.
	`'*****************************************************************************`
	`'**** Read Plan Time (Power of machine is on)                                 ****`
	`'*****************************************************************************`
	`strSQL = "SELECT * FROM AlgHitListResult_tmp WHERE MsgNr='" & lngMsgIDPlan & "'"`
	`objCommand.CommandText = strSQL`
	`Set objRecordset = objCommand.Execute`
	
	`lngCount = objRecordset.Fields.Count`
	
	`If (lngCount>0) Then`
	`  objRecordset.movefirst`
	`  tmpPlan = objRecordset.Fields(41).Value`
	`Else`
	`  HMIRuntime.Trace "Selection returned no fields" & vbNewLine`
	`End If`
	To be able to read the utilization time, the breakdown time and the number of failures, there will be further database inquiries. To do this, each "SELECT" instruction with the respective message ID is stored in the "strSQL" string tag and transferred to the "Command" object.

5.	The detected values are stored in the following temporary VB tags: • "tmpPlan" plan holding time in seconds • "tmpUtil" utilization time in seconds • "tmpTecDis" breakdown time in seconds • "tmpFCount" number of failures

6.	Once the data is read out the database connection is closed again.
	`'*****************************************************************************`
	`'**** Close database connection                                              ****`
	`'*****************************************************************************`
	`objConnection.Close`
	
	`Set objCommand = Nothing`
	`Set objRecordset = Nothing`
	`Set objConnection = Nothing`

4.4.2 KPI calculation

Table 4-9

No.	Instruction

1.	The screen shows the tag declaration of the WinCC tag objects and the temporary VBS tags for calculating the production figures.
	`'*****************************************************************************`
	`'**** Objects for WinCC tags                                                    ****`
	`'*****************************************************************************`
	`Dim T_Plan, T_Util, T_TecDis`
	`Dim MTBF, MTTR`
	`Dim Utilization, Availability`
	
	`'*****************************************************************************`
	`'**** Temporary tags for KPI calculation                                        ****`
	`'*****************************************************************************`
	`Dim tmpPlan, tmpUtil, tmpTecDis`
	`Dim tmpMTBF, tmpMTTR`
	`Dim tmpFCount`

2.	The calculation of the KPI is indicated by the following terms:
	• Availability
	$Availability = \frac{T_{Plan} - T_{TecDis}}{T_{Plan}} \cdot 100$
	• Utilization
	$Utilization = \frac{T_{Util}}{T_{Plan}} \cdot 100$
	• Mean time between failures
	$MTBF = \frac{T_{Util}}{[Counts\ of\ Failure]}$
	• Mean time to recover
	$MTTR = \frac{T_{TecDis}}{[Counts\ of\ Failure]}$
	
	`'*****************************************************************************`
	`'**** Calculation of Key Performance Indicator                                ****`
	`'*****************************************************************************`
	`Utilization.Value = (tmpUtil / tmpPlan) * 100`
	`Availability.Value = ((tmpPlan - tmpTecDis) / tmpPlan) * 100`
	`tmpMTBF = tmpUtil / tmpFCount`
	`tmpMTTR = tmpTecDis / tmpFCount`

4.4.3 Conversion of second values

Times are stored in the database in second values. To make these details easier to read for the user, they are converted into "HH:MM:SS" format and stored as a string in the WinCC tag.

Table 4-10

No.	Instruction

1.	The following tags were declared for the conversion:
	`'*****************************************************************************`
	`'**** Temporary tags for changing DB-value in readable time format  ****`
	`'*****************************************************************************`
	`Dim h,m,s,sH,sM,sS`

2.	On the example of plan holding time, this code section shows the conversion of the second value read from the database into "HH:MM:SS" format:
	`'*****************************************************************************`
	`'**** Output Values in WinCC                                                    ****`
	`'*****************************************************************************`
	`' Convert seconds to time-string (Plan - time)`
	`h = Int(tmpPlan / 3600)`
	`m = Int(tmpPlan / 60) Mod 60`
	`s = tmpPlan Mod 60`
	`If h<10 Then sH = CStr("0" & h) Else sH = CStr(h)`
	`If m<10 Then sM = CStr("0" & m) Else sM = CStr(m)`
	`If s<10 Then sS = CStr("0" & s) Else sS = CStr(s)`
	`T_Plan.Value = sh & ":" & sM & ":" & sS`

3.	Once all calculations are finished, all values are written into the respective WinCC tags:
	`'*****************************************************************************`
	`'**** Write output values to WinCC tags                                          ****`
	`'*****************************************************************************`
	`T_Util.Write  'Utilization time`
	`T_Plan.Write  'Plan holding time`
	`T_TecDis.Write  'Brakdown time`
	`Utilization.Write  'Utilization`
	`Availability.Write  'Availability`
	`MTBF.Write  'Mean time between failure`
	`MTTR.Write  'Mean time to recover`

5 Startup of the application

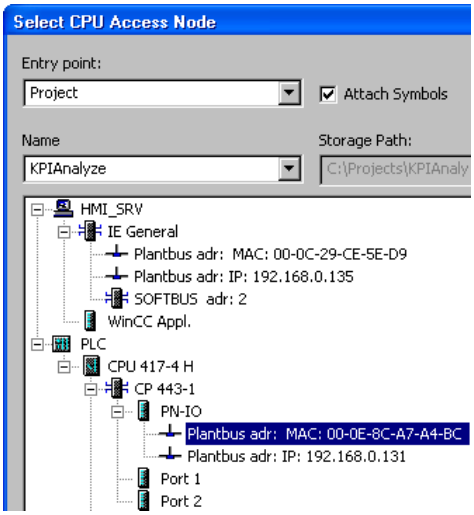
5.1 Preparation

Table 5-11

No.	Instruction	Comment
1	Unzip the "WinCC_KPI_Analyze_v10.zip" file included in delivery using the menu command "File > Retrieve…" of the SIMATIC Manager.	

5.2 Startup

Table 5-12

No.	Instruction	Comment
1	Open the "KPIAnalyze" project in the SIMATIC Manager.	
2	Start the "S7-PLCSIM" program and select the MAC address of the S7 station from the "Select CPU Access Node" dialog. The WinCC project is setup for this connecion. Alternatively you can load the program also into an existing S7 station. For this purpose, the hardware and the connections have to be configured accordingly.	
3	Load the S7 program into the simulator.	
4	Open the OS project and start WinCC Runtime.	If you are using a different connection or another S7 station, the OS project has to be recompiled and the connection settings in WinCC have to be adjusted.

6 Operation of the Application

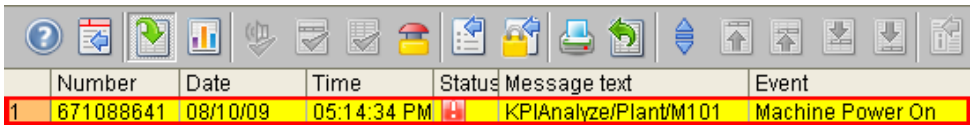
6.1 Recording the times

The times which are used as basis for calculating the production figures are recorded via the switches "POWER" (plan holding time), "START" (utilization time) and "Generate FAILURE" (breakdown time).

6.1.1 Plan holding time

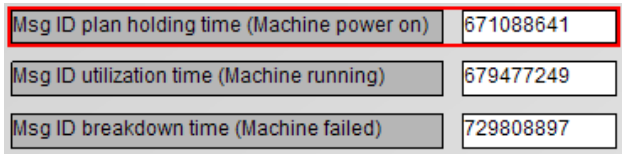
The plan holding time is recorded has soon as the message "Machine Power On" is pending.

Figure 6-12



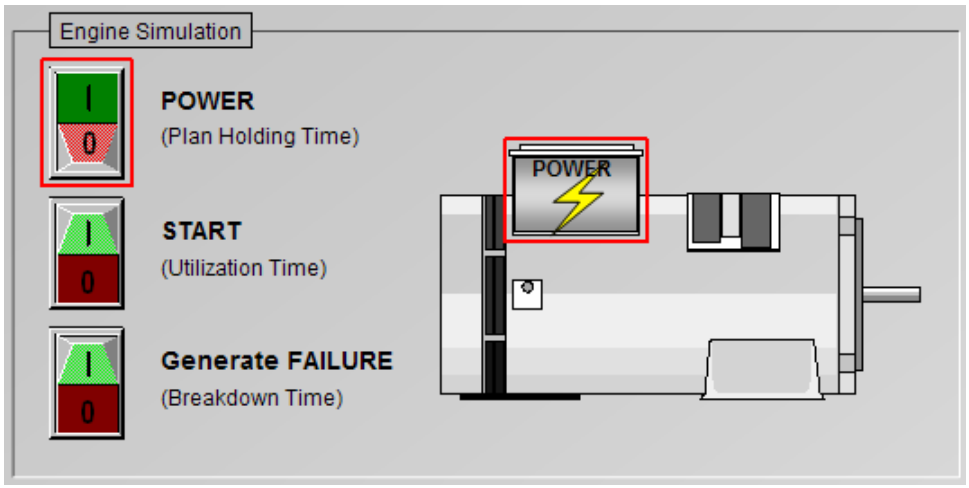
To be able to calculate the KPI, enter the message ID of the "Machine Power On" message into the "Msg ID plan holding time" input field.

Figure 6-13



The message is triggered by the "POWER" switch. The operation of simulated motor is shown by the lightening icon on the motor icon.

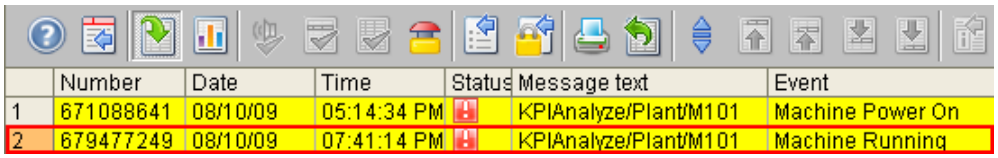
Figure 6-14



6.1.2 Utilization time

The utilization time is recorded has soon as the messages "Machine Power On" and "Machine Running" are pending.

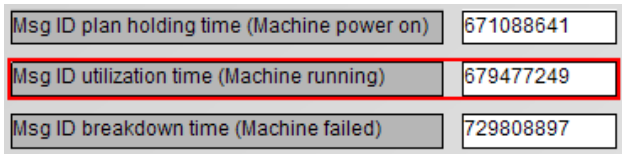
Figure 6-15



	Number	Date	Time	Status	Message text	Event
1	671088641	08/10/09	05:14:34 PM		KPIAnalyze/Plant/M101	Machine Power On
2	679477249	08/10/09	07:41:14 PM		KPIAnalyze/Plant/M101	Machine Running

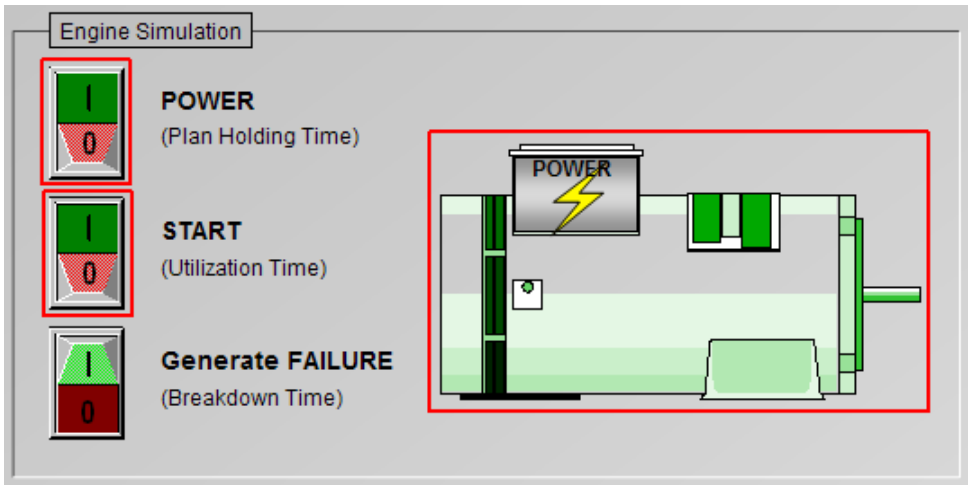
To be able to calculate the KPI, enter the message ID of the "Machine Running" message into the "Msg ID utilization time" input field.

Figure 6-16



The message "Machine Running" is triggered by the "START" switch. The green color of the motor icon shows that the motor is running.

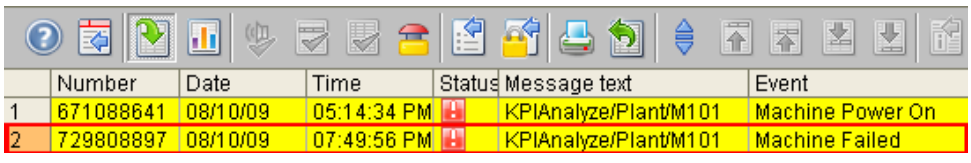
Figure 6-17



6.1.3 Breakdown time

The breakdown time is recorded has soon as the messages "Machine Power On" and "Machine Failed" are pending. The states "Machine Running" and "Machine Failed" cannot occur at the same time.

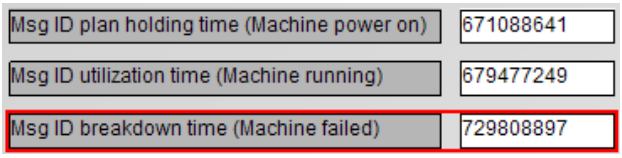
Figure 6-18



	Number	Date	Time	Status	Message text	Event
1	671088641	08/10/09	05:14:34 PM		KPIAnalyze/Plant/M101	Machine Power On
2	729808897	08/10/09	07:49:56 PM		KPIAnalyze/Plant/M101	Machine Failed

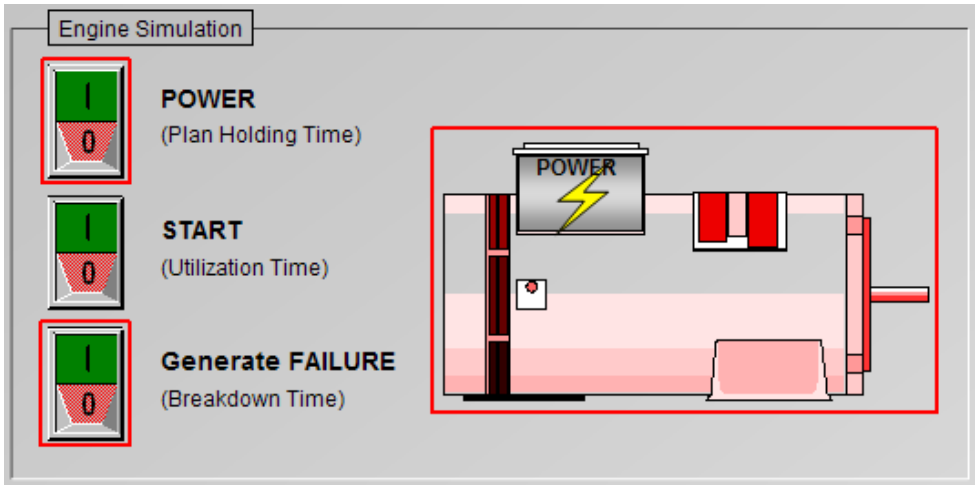
To be able to calculate the KPI, enter the message ID of the "Machine Failed" message into the "Msg ID breakdown time" input field.

Figure 6-19



The message "Machine Failed" is triggered by the "Generate FAILURE" switch. It is irrelevant whether the "START" switch is on or off. The error state is indicated by a red flashing motor icon.

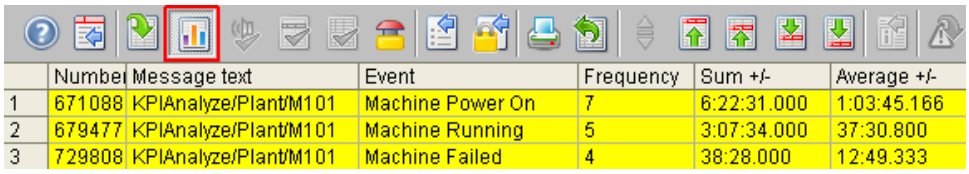
Figure 6-20



6.2 Updating database

The hit list of the WinCC AlarmControl carries out statistical calculations of the alarm messages and saves them temporarily in the WinCC Runtime database. For the temporary database table to be updated it is necessary to either change the screen or to go to the WinCC AlarmControl (e.g. message list-> hit list) view.

Figure 6-21



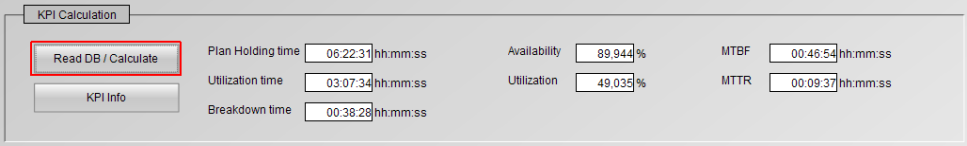
	Number	Message text	Event	Frequency	Sum +/-	Average +/-
1	671088	KPIAnalyze/Plant/M101	Machine Power On	7	6:22:31.000	1:03:45.166
2	679477	KPIAnalyze/Plant/M101	Machine Running	5	3:07:34.000	37:30.800
3	729808	KPIAnalyze/Plant/M101	Machine Failed	4	38:28.000	12:49.333

CAUTION The time how long a message was pending can only be recorded when it disappears again. For this reason, to avoid arithmetic errors, there should not be any messages pending in the message list before calling the hit list.

6.3 KPI calculation

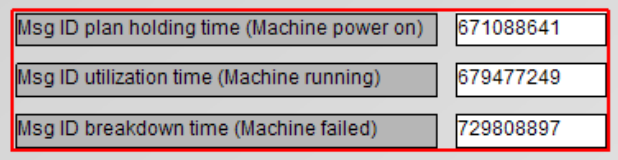
Press the "Read DB / Calculate" button to initiate the calculation of the KPI. The detected and calculated values are displayed in the respective output fields.

Figure 6-22



Make sure that the message IDs for the database inquiries are entered correctly. Otherwise the database could not be read out correctly or the calculation of the values would be faulty.

Figure 6-23



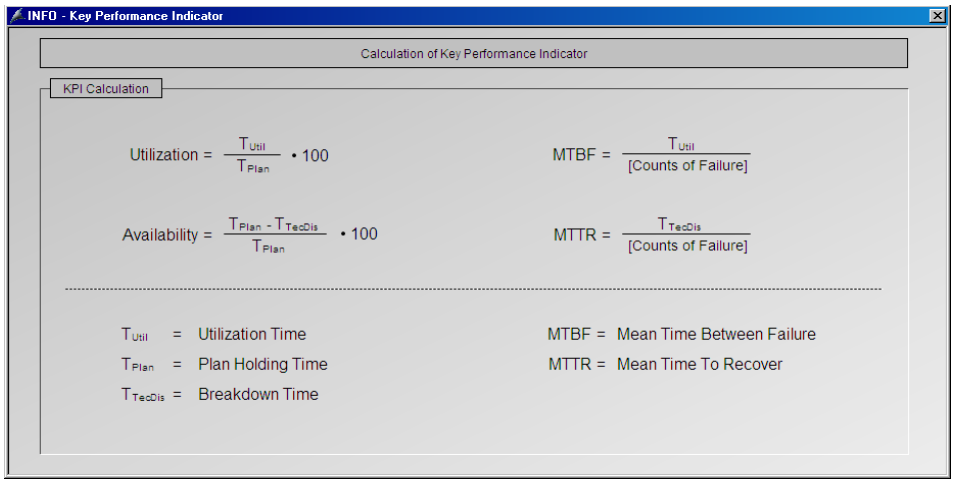
The following values are read out of the database table and are used for the calculations:

Table 6-24

Field	Message	Description
Sum +/-	"Machine Power On"	The total of all the times, from the arrival of the message until it disappears again. This time corresponds to the plan holding time.
Sum +/-	"Machine Running"	The total of all the times, from the arrival of the message until it disappears again. This time corresponds to the utilization time.
Sum +/-	"Machine Failed"	The total of all the times, from the arrival of the message until it disappears again. This time corresponds to the breakdown time.
Frequency	"Machine Failed"	The frequeny of messages with this status. This value corresponds to the frequency of failures.

To be able to receive information on the terms used in this example, press the "KPI Info" button. A screen will appear which displays the "KPIInfo.pdl" WinCC screen.

Figure 6-25



7 Glossary

Key Performance Indicator (KPI)

Describes the figures which are used for recording and optimizing production processes.

MTBF (Mean Time Between Failures)

Describes the average operating time between the failures of a systems

MTTR (Mean Time To Recover)

Describes the average time for recovery after the failure of a system.

8 Related Literature

8.1 Internet Links

This list is not complete and only represents a selection of relevant information.

Table 8-26 Internet links

	Topic	Title
\1\	Reference to the entry	http://support.automation.siemens.com/WW/view/en/EntryID
\2\	Siemens I IA/DT Customer Support	http://support.automation.siemens.com
\3\	FAQ	How to use message classes if WinCC is integrated in the STEP 7 project http://support.automation.siemens.com/WW/view/en/31622970
\4\	SIMATIC WinCC/Connectivity Pack	http://support.automation.siemens.com/WW/view/en/28887620
\5\	SIMATIC Maintenance Station	http://support.automation.siemens.com/WW/view/en/31238198
\6\	SIMATIC WinCC/Downtime Monitor	http://support.automation.siemens.com/WW/view/en/34519922

9 History

Table 9-27 History

Version	Date	Modifications
V1.0	01.12.2009	First version