## **SPS-Sensors for X-Monitors**

(SpsEyes - SpsGold - SpsOrion - SpsPrtg)



# White Paper

#### Contents

C	Copyright	
1.	Overview	
Р	Prerequisites	
S	Supported Systems	
Т	The Working	
2.	The SPS-Profiler	5
D	Definition Access for SPS	5
Iı	nteractive Testing	
3.	The SPS-Ranges	
4.	The SPS Monitoring	
5.	Monitoring with SpsEyes	13

#### **Copyright**

The Copyright 2014-17 for the SPS-Sensors is owned by ICON Software GmbH.

## ICON Software GmbH Am Dorfplatz 10 D-92540 Altendorf

#### Telefon: 09675/9134-00

Web: <u>http://www.icon-software.de/</u> Mail: <u>Info@ICON-Software.de</u>

Used Trademarks:

SIMATIC® ist ein eingetragenes Warenzeichen der Siemens AG PRTG, Paessler sind eingetragene Warenzeichen der Paessler AG Windows® ist ein eingetragenes Warenzeichen der Microsoft Corporation

The used productnames and trademarks are only for identification and could be registrated trademarks of the maufacturer

Technical changes reserved

## 1. <u>Overview</u>

With the help of the SPS-Sensors you are able to get a lot of informations from Siemens S7 SPSs and direct them to the Network Monitor PRTG from PAESSLER. Therefore no intervention to the SPS-System is required. The access to the SPS-Systems is done via the S7-Protocoll by TCP/IP and requires no intervention within the SPS. For the access to the SPS the IP-Address and the Rack- and Slot-Number is required.

#### **Prerequisites**

For the usage of SPS-Sensors you need:

- The sensors are to put into the "...\ExeXml"-directory for PRTG
- You need the file "**SpsSnr.Dll**". Therefore you have to put the file into the "…\ExeXml"-directory of PRTG too
- For working the sensors you have to create a profile for accessing SPS. You can create it with the tool "SPS Profiler.Exe" which is part of the installation.

#### **Supported Systems**

Following SPSs are supported:

	Siemens	S7-200 S7-400 S7-1500	S7-300 S7-1200 !Logo
-	with CP (C	P242-1, CP343-1, <b>G</b>	CP443-1,LEAN)

- VIPA S7 as all S7-compatible systems

## **The Working**

For accessing a SPS-System a profile with the informations about the accessing-data like IP-Address, Rack- and Slot-Number is needed. This profile is generated via the tool "**SPS Profiler.Exe**" which is part of the installation. The profile is stored into the Xml-Directory of PRTG. With the Profiler you can directly contact the SPS and view the informations.



First, the sensor reads out the profile and opens a connection to SPS via RFC1006. With the knowledge of IP-Adresss, the Rack- and the Slot-Number the SPS could be accessed. Then the required function is processed within SPS and the resulting is transferred back to the sensor. The sensor formats it and divides it to an information-line and channels. This data are transferred then to PRTG.

Within the PRTG-System the delivered informations are collected and prepared for analysis. They could be shown graphically and used for generating a mail or SMS to notify an administrator.

## 2. <u>The SPS-Profiler</u>

For the usage of sensors there is the definition of a profile required. You could create a profile via the "**SPS Profiler.Exe**" and put in the necessary informations for connecting to SPS. Additionally you are able to view the resulting before you define it within PRTG.

## **Definition Access for SPS**

\_

For accessing a SPS System you need a profile. The nessesary informations are entered via the Profiler. The profile itself will be stored in the PRTG-Directory ,,...\ExeXml".

files	Rack/Sit: S7 - 300 / 00 / 02	C State's
	(S00.02) Descript.: S7-313 Kühlhaus	C System
tails	SPS: 10.96.230.31 Alias: Sps05 R	ack / Slot / Lang
	Type.: S7-300 👻 Pass: 0	0 \$02 \$ DE •
o's	Info: No Info available / CPU Info failed 🤅	Pg C Op C S7
sc's	Desc.: S7-313 Kühlhaus	Async Accesses
o's	Info: No Info available / CPU Info failed ( Desc.: S7-313 Kühlhaus	Pg O Op O Async Acce

The name of the profile is to be used by definition a new sensor within PRTG. The Profile-Name has to be the first parameter at the PRTG-Sensor.

For verify tha access the following functions are supported:

- "**Ping**": Ping the SPS (Function of the operating system is used)
  - "Info": Reading the general informations about the SPS System
- "Check": Verify the hardware (SPS-Model, and so on)

With the following steps you are able to define a new profile.

Step 1: The access to the SPS is checked by the method "Ping":



Step 2: With the "Info" connection informations a read from the SPS-System:

ofiles	Sps04	Sps/Alias: 10.96.230.31 / S Rack/Slt: S7 - 300 / 00 / 00 Sps-Info.: (No AS-Name)	Sps04 C Basics D2 C State's C Values
	(.\$00.02)	Descript.: S7-312C Hebebü	ühne 🧿 System
2	x cDC Catala	- Destinate OK	
etails	* SPS Catalo	g: Reading is UN Code: 6ES7 315-2EH14-0AI	480
1	* SPS Version	n 32.9.9	
	* Max PDU L	enath: 2048 Butes	
ifo's	* Max Conne	ctions.: 1024 Users	
3	* Max MPI R * Max Bus Ra	ate: 12000 K-Bps ate: 187 K-Bps	
1000	SPS Info	was successfully	
esc s	THE OT OT INTO	Contraction and the second second second	

Step 3: With the "Check"-Function general information about the SPS are read:

SPS-Profile	r for Simatic S7
Profiles	Sps/Alias:     10.96.230.31 / Sps04     C     Basics       Rack/Sit:     S7 · 300 / 00 / 02     C     State's       Sps04     Sps-Info;     (No AS-Name)     C     Values       (s00.02)     Descript;     S7-312C Hebebühne     C     System
Details Info's	* Module Type Name: Cpu 312C * Serial Number: S C-C9Tb25742012 * AS Name: Simatic-Station * Module Name: Cpu 312C * SPS-Copyright: Original Siemens Equipment * PLC-Date/Time: 15.03.15 / 08.44 * SIc-Protection Level: Can Read * Prm-Protection Level: No Password
Desc's	🐺 SPS Check was successfully
Ø	Save Ping Info Done Trace
Back	Copyright 2011-16 ICON Software GmbH, All Rights reserved, Ver 4.2 Icon

After the creation of the profile you can use the sensors, for example the basis-functions:

files SpsDemo (.S00.02)	ps/Alias: 10.96.xx.xx / SpsDemo lack/Slt: S7 - 300 / 00 / 02 ps-Info.: Simatic-Server lescript.: SPS Samples	Basics State's Values System
tails PSPS LedSta SPS MemU: SPS SysAre SPS SysBic SPS SysInfo SPS SysChe	tes Listing of the LED-States sag Listing of Memory-Usages as Listing of System-Areas ks List maximal Block-Entries Informations and the Response checking for the CP and CPU	Available Available Available Available Available Available
c's 📴 SPS SysChe	ck SpsDemo.S (No Parame	ters available)

With the "Exec"-Button you release the access:

	Sps/Alias: 10.96.xx.xx / Sps/	Demo	Basics
Profiles	Rack/Sit: 57 - 300 / 00 / 02		C State's
	(S00.02) Descript.; SPS Samples	1.1	C System
Details	[SPS] S7-312C Lifting Station System Check	ing for SPS-Sta	ite .
	🐹 Chn.01: PLC-State Running	1	State
E.	🔜 Chn.02: PLC-State Stopped	0	State
Info's	🔜 Chn.03: PLC-State UnKnown	0	State
	Physical Chr. 04: Response-Time	25	mSec
Desc's	📰 SPS SysCheck 🛛 📗 SpsDemo.S	(No Paramet	ers available)
Desc's	SPS SysCheck SpsDema S	(No Paramet	ers available

With the "Apply"-Button you get the needed information here for PRTG:

	Sps/Alias: 10.96.xx.xx / SpsDemo	Basics
Profiles	Rack/Sit: 57-300700702	C State's
Parameters.	SpsDemo Sps-Info.: Simatic-Server	C Values
	Lisuuuzi Descript.; SPS Samples	C System
Detaile	Sensor-Definition within PRTG, User-Defined Sensor	r, Exec/Scripts
Details		
	Sensor Checking for the CP and CPU.	within PRTG
E	🛛 🔀 Exec "SPS SysCheck.Exe"	within PRTG
Info's	Param(s) SpsDemo.S00.02	within PRTG
e	Mutex: SosPrtg (Recommended)	within PRTG
)esc's	SPS SysCheck SpsDemo.S (No F	<sup>o</sup> arameters available)
123	Defet a la Charle D	wash trace

#### **Interactive Testing**

By using the "SPS Profiler" you have the possibility to test the sensors interactively before implementing them to PRTG. There are following ranges:

-	Basics	Functions from the BASICS
-	States	Functions for the range STATE
-	Values	Show the fields with a value
-	System	Reading System-Informations

Within the range of SPS-Basis general informations about the basics is offered. For example, this are informations about Versions, or about the used CPU.

The range "States" is to watch the states of the SPS . For example you see the input or output flags.

By the "Values" you are able to read out the integrated values with are stored within the SPS.

With the range "System" you get informations about the environment of the SPS.

Example for the usage of the Sensor ,,ListSzlGrps":



This example shows the available groups within the SZL "0".

Sample for the sensor "SPS Directory":



Shown are the available entries for the stored blocks.

Sample for the sensor "SPS SysStates":

øj	Sp Sp	s/Alias: 10.96.xx.xx / Sp	isDemo	C Ba	sics
Profiles	CarDana So	CK/SIC S7 - 300 / 00 / 1	J2	(• Sta	ite's
5	(.S00.02) De	escript.: SPS Samples		C Sys	tem
Details	(SPS) S7-312C Li	fting Station Values of AB	3=0 / Platform Act	ivities	1
	🗸 Chn.01: A0.	0: Power Supply	1	On/Off	
E C	🗸 Chn.02: A0.	1: Upwards	0	Moving	E
Info's	🗸 Chn.03: A0.	2: Downwards	0	Moving	
	🔜 Chn.04: A0.	3: Ready for Up	1	On/Off	×
Desc's	📕 SPS SysState	s 🎼 SpsDemo.S	Output AB=0	(	

Shows are the states for the output group "0".

## 3. The SPS-Ranges

An overview of the available functions is shown with the interactively presentation by the Profiler. This is an selfaxtracting file who shows all the available functions by examples. The file could be downloaded at:

#### www.ICON-Software.de/SpsDemo.exe

The range "Basics" shows the general functions:



The range "States" shows all the available states:

![](_page_9_Picture_6.jpeg)

The range "Values" shows the available values:

![](_page_10_Picture_1.jpeg)

The range "System" shows the internal informations of the SPS:

![](_page_10_Picture_3.jpeg)

The result of every of the sensors could be show by pressing the "Exec"-Button. This is is done before the sensor is planned within the PRTG

#### 4. The SPS Monitoring

The SPS-Sensors could be directly imbedded to the available monitoring-systems, here the PRTG from Paessler. With the usage of parameters you could select all the values you want to monitor.

Within the monitoring only the assignment with the attached sensor is done. Then you are able to define the time interval and the reactions in case of errors.

E1.1: Sensor_1			E1.2: Sensor_2 1 State	~
	1		E1.3: Sensor_3 0 State	0
) State	0	1 State	E2.1: Sensor_1 1 State	3
2.2: Sensor_2	E2.3: Sensor_3 1 State	2	E3.1: Sensor_1 1 State	3
3.2: Sensor_2	E3.3: Sensor_3			

This sample shows the working of three different sensors.

![](_page_11_Picture_5.jpeg)

This sample shows the states of the input group "EB1".

![](_page_11_Picture_7.jpeg)

This sample shows the states of the output group AB1".

## 5. <u>Monitoring with SpsEyes</u>

Beside the Monitoring-Systems like PRTG, Orion und WhatsUp Gold the SPS-Sensoren could be used with SpsEyes too. SpsEyes is a separate software to monitor by usage the sensors directly.

Example: Layout for monitorng a single SPS-Device:

![](_page_12_Picture_3.jpeg)

Example: Alternate layout for monitoring a single SPS-Device:

	Der-Los	22			P	P	0
Ides State		Li Cando	21	So State	Sec Traffic	Nec Ourb	Netleti 42
System SD Care	User-CPU 5 %	CPU-Unge 0 %5	Mes.Pres 1545 Millions	Pages Used	Raft Used	Fagnila	Pager Out
CPO-Osages (%)	1 mar		-	-	-	The second	2
Disk-Usages (%)	t t zeris)	n	115 Londs	242 Lands	185 Leads	34	+0
0 14:02 14:04 14:07	ante Time						
		SE	JOB	306	JOE	JOE	JUE Deak John
		Act Update		11.14		10	ist.
200	Response RFC-Open	E	50	Parting(i)	Facting(i)	Sport Err	Spost Err

Every of this objects works autonomous, every object is realized by an own process. Every process works independent from other processes and communicates via RFC with the attached SPS-System.

Example: Layout for monitoring with more SPS-Systems:

![](_page_13_Picture_1.jpeg)

On the left hand general information's like CPU-, Disk-Activities or the number of Users are presented via Graphics or Gauges.

On the right side a matrix organized by SPS-Systems and functions is shown. There are three SPS-Systems and functions like Cpu-Times, Data sent and so on.

Remarks:

- By Graphics, a single value is requested from each SPS-System. The values will be prepared and analysed
- By Gauges, a single value is requested and shown
- By Images the values of the sensors are analysed and presented by colours

All the sensors could be modified by size, layout and position during runtime. New sensors could be added at any time. You are able to design your own Dashboard.