# 如何在 WinCC flexible 操作面板上显示 CPU 的当前状态?

# WinCC flexible

# FAQ • June 2008



# Service & Support

Answers for industry.

# **SIEMENS**

本条目来自 Service & Support Portal of Siemens AG, Sector Industry, Industry Automation and Drive Technologies。使用的环境遵循 (www.siemens.com/nutzungsbedingungen).

从下列链接下载本文档。

http://support.automation.siemens.com/WW/view/de/35538370

问题

如何在 WinCC flexible 操作面板上显示 CPU 的当前状态?

答案

文档中的描述和条目详尽地回答了上面的问题。

1

# 目录

1	WinCC flexible 2008 及之前版本步骤	4
1.1	介绍	Error! Bookmark not defined.
1.2	支持的操作面板	
1.3	STEP 7 中的设置/组态	
1.4	WinCC flexible 中的组态	
1.4.1	定义区域指针	7
1.4.2	创建变量和状态显示	7
1.4.3	组态消息窗口	
1.4.4	组态调度和创建状态显示	
2	WinCC flexible 2008 SP1 中的步骤	
2.1	介绍	Error! Bookmark not defined.
2.2	支持的操作面板	
2.3	显示"System Messages" 编辑器	
2.4	组态CPU 状态	

### 1 WinCC flexible 2008 及之前版本步骤

### 1.1 介绍

本 FAQ 组态可以可靠地显示 CPU 状态。用 I/O 域或系统信息来显示控制器 状态的传统方式并不非常的可靠。下面的组态可以显示三种状态。

- CPU"运行"
- CPU"停止"
- 没有连接到 CPU

#### CPU "运行"

图 1-1



使用 clock memory,可以检查 CPU 是否处于"运行"状态。只要面板上可 以显示 clock memory,就可以断定是"运行"状态。

#### CPU"停止"

图 1-2



"协调"区域指针可以表示操作员面板正和 控制器通讯。

图 1-3



操作员面板以大约1秒的频率翻转 life bit。在控制器程序中通过查讯该位来 判断是否与操作员面板有连接,就可以判断 CPU 仅仅是处于"停止"模式 还是连接无效。

#### 没有连接到 CPU (断电)

图 1-4



系统消息提供了操作员面板和控制器的运行模式信息。系统消息 140000 ("已建立与 PLC 的连接") 和 140001 ("已断开与 PLC 的连接") 显示了操作员面板 是否连接到 CPU。

#### 1.2 支持的操作面板

支持 "协调"区域指针的所有面板。

**注释** OP 73micro, TP 170micro, TP 177micro and TP 170A 不支持 "协调"区域 指针。

#### **1.3** STEP 7 中的设置/组态

在硬件组态中设置 clock memory

图 1-4

General	Startup Cyc	cle/Clock Memory Retentive Memory In	terrupt
Cycle			
IØ Update	e OB1 process image cyc		
Scan cycle	e monitoring time [ms]:	150	
Minimum s	can cycle time (ms)	0	
Scan cycle	e load from communication	on [%]: 20	
Size of the	process image:	×	
OB85 · cal	I up at I/O access error:	No OB85 call up	-
Clock Men	nory		
Clock	memory		
Memoru Bi	ite:	(10)	

在面板上,通过 clock memory 就可以判断 CPU 是否处于"运行"状态。 只有 CPU 处于"运行"模式. clock memory 才可以到达操作员面板。

在 STEP 7 中不需要其他设置。

在符号表中,为"协调"区域指针设置标记词。

图 1-5

a s	7-Program	mm(3) (Symbole	00(2)\CPU 315-2 DP			
	Status	Oynibol -	Adresse	Duloniyp	Hommonica	
1	-	Koordinierung	M/V 12	WORD	Bereichszeiger_Koordinierung	
2		Taktmerker	MB 10	BYTE	CPU Taktmerker	
-						

#### 1.4 WinCC flexible 中的配置

打开 WinCC flexible ES,若是集成项目,会自动创建到 CPU 的连接。若是 非集成项目,创建一个连接。

#### 1.4.1 定义区域指针

激活 "协调" 区域指针,设置对应的地址。

图 1-6

					CON	NECLIC	NS
Name	Active	Communication driver	Station	Partner	Node	Online	Comme
Verbindung_1	On	▼ SIMATIC S7 300/400	▼ \status_sps\S	CPU 315-2 DP	▼ CP 343-1	▼ On	-
		<					>
Parameters	Area pointer						
or all connections							
Connection	Name	Symbol	Address		Length	Trigger mode	Acq
<undefined></undefined>	Screen number	<undefined></undefined>			5	Cyclic continuous	<un< td=""></un<>
<undefined></undefined>	Date/time PLC	<undefined></undefined>			6	Cyclic continuous	<un< td=""></un<>
<undefined></undefined>	Project ID	<undefined></undefined>			1	Cyclic continuous	<un< td=""></un<>
	<	W. I					>
or each connectio	n						
Active	Name	Symbol	Address		Length	Trigger mode	Acqu
Off	Data record	<undefined></undefined>			5	Cyclic continuous	<un< td=""></un<>
Off	Date/time	<undefined></undefined>			6	Cyclic continuous	<un< td=""></un<>
On	Coordination	Koordinierung	MW 12		1	Cyclic continuous	<un< td=""></un<>
Off	Job mailbox	<undefined></undefined>			4	Cyclic continuous	<un< td=""></un<>
_	<	ar I					>

#### 1.4.2 创建变量和状态显示

在通讯>变量下创建下列变量:

- Run\_error (内部变量, ULong)
- Online\_error (内部变量, ULong)
- Coordination (Connection\_1, Word, MW 12)
- Clock memory (Connection\_1, Byte, MB 10)

#### 图 1-7

						V	AR	TABLEN
1	ame	Anzeigename	Verbindung 🔺	Datentyp		Symbol		Adresse
	RunFehler		<interne td="" vari="" 💌<=""><td>ULong</td><td>-</td><td><undefiniert></undefiniert></td><td>-</td><td><keine adresse=""></keine></td></interne>	ULong	-	<undefiniert></undefiniert>	-	<keine adresse=""></keine>
	OnlineFehler		<interne variable=""></interne>	ULong		<undefiniert></undefiniert>		<keine adresse=""></keine>
	Koordini <del>c</del> rung		Verbindung_1	Word		Koordinierung		MW 12
	Taktmerker		Verbindung_1	Byte		Taktmerker		MB 10

• 为 "Coordination" 和 "Clock Memory" 变量配置下列事件。

#### Coordination

 在事件 > 更改值下,对内部变量 "Online\_error" 置 0,对内部变量 "Run\_error" 加 1。

图 1-8

					TA	35
Name Dis	play nami	e Connection	Data type	Symbol	Address	Street Street
RunFehler		<internal tag=""></internal>	ULong	<undefined></undefined>	<no address=""></no>	1000
OnlineFehler		<internal tag=""></internal>	ULong	<undefined></undefined>	<no address=""></no>	
Koordinierung		Verbindung_1	Word	Koordinierung	MW 12	3
Taktmerker		Verbindung_1	Byte	Taktmerker	MB 10	
<						3
ordinierung (T	ag)					90
General Properties Events	ag)   🔀 4   1	▶ ↓ III D SetValue			Function I	۹( List
General Properties Events Change value	ag)   <u>X</u>    1			CPU\OnlineFehler	Function I	9( List
General Properties Events Change value High limit Low limit	ag) X1	Image: SetValue       Tag (Out)       Value		CPU\OnlineFehler 0	Function I	۹( List ⊻
General Properties Events Change value High limit Low limit	ag)			CPU\OnlineFehler 0	Function I	۹( List
General Properties Events Change value High limit Low limit	ag) 1 1 2	► ● E B SetValue Tag (Out) Value E IncreaseValue Tag (InCut)		CPU\OnlineFehler 0 CPU\RunFehler	Function I	¶( List ⊻

#### **Clock memory**

 在事件 > 更改值下,对内部变量 "Online\_error" 置 0,对内部变量 "Run\_error" 置 0。

图 1-9

Name D	isplay nam	e Connection 4	Data type	Symbol	TAG Address
RunFehler		<internal tag=""></internal>	ULong	<undefined></undefined>	<no address=""></no>
OnlineFehler		<internal tag=""></internal>	ULong	<undefined></undefined>	<no address=""></no>
Koordinierung		Verbindung_1	Word	Koordinierung	MW 12
Taktmerker		Verbindung_1	Byte	Taktmerker	MB 10
roperties Events					1 001001011 00
ivents Change value	1	🗄 Set¥alue			
<ul> <li>High limit</li> </ul>		Tag (Out)		CPU\RunFehler	
<ul> <li>Low limit</li> </ul>		Value		0	
	2	🗆 Set¥alue			
	2	Tag (Out)		CPU\OnlineFehler	

该配置使内部变量 "Run\_error" 大概每秒置 1, 然后 CPU 的 clock memory 立刻使之复位。

当"Run\_error"大于等于1时,表明 CPU 处于"停止"模式。根据下图设置 动画 "Visibility"。

#### 图 1-10

Vo	rlage TextField	Ston (Text Field)	
-	General Properties Animations Appearance	✓ Enabled      Tag     Object state	Visibility
	Diagonal Movement     Horizontal Movement     Vertical movement     Direct Movement     Vietbility	Run_Fehler Visible	
	VISIONICY	Integer     Range from 0      And And And And And And And And And	to 1

在层1创建一个不带动画的文本域为"RUN"状态。该文本域处于前端。直到 其它状态被激活。

	冬	·	1-	1	1		
k	CF	ગ	J	R	u	n	Ī

注释 可以灵活组态显示方式,也可以不用文本域显示。但是要保证显示对象的 动画组态相同。

#### 1.4.3 组态消息窗口

按照下图组态消息窗口。在下一章节,在调度器的作业中可以组态打开和关 闭消息窗口对话框。可以通过 DialogOpen 和 DialogClose 判断 CPU 是否 在线。

图 1-12

Syste	m	×	•
	, Uhrzeit Datum		
n Vorlage Meldel	n fenster (Alarm Window)		<b>Q</b> (
General     Properties	- Disalau		General
Animations	<ul> <li>Orspray</li> <li>Orspray</li></ul>	Alarm classes	F
	Unacknowledged alarms     Alarm events	Diagnosis Events Warnings	
	Alarm log	System	ম

#### 1.4.4 组态调度器和创建状态显示

在 "调度器" 中创建 3 个作业, 配置如下图。

- DialogClose
- DialogOpen
- Cycl\_1min

#### 图 1-13

		SCHEDULE
Name	Event	Description
DialogClose	When dialog is closed	Perform when a modal dialog is closed.
DialogOpen	When dialog is opened	Perform when a modal dialog is opened.
Zykl_1min	1 Minute	Perform every minute.

#### DialogClose

在作业栏中组态 "Increase Value" 功能,链接到内部变量 "Online\_error" (Online\_Fehler). 其值被加 1。

图 1-14

						SCHEDULE:	2
Name		Event	33	Description		Comment	
DialogClose When dialog is closed		l .	Perform when a modal dialog is closed.				
		<	•		11.4.1 × 1	2	Č
Job			×	++ ==		Function list	t
Name	Name DialogClose		1	🗆 Increase	/alue		-
Event When dialog is closed		sed 👻		Tag (InO	ut)	CPU\Online_Fehler	
				Value		1	
Perform when a modal dialog is closed.		2	<no funct<="" td=""><td>ion&gt;</td><td></td><td></td></no>	ion>			

#### DialogOpen

在作业栏中组态 "Increase Value" 功能, 链接到内部变量 "Online\_error" (Online\_Fehler). 其值被加 1。

图 1-15

Name	Event		Description	Comment	
DialogClose When dialog is closed		d	Perform when a modal dialog is cl	osed.	
DialogOpen	When dialog is open	ed	Perform when a modal dialog is o	pened.	
	<				>
Job		×	+ + <b>Ⅲ</b>	Fun	ction lis
Name	DialogOpen	1	Increase¥alue		
Event	When dialog is opened	Ī	Tag (InOut)	CPU\Online_Fehler	
			Value	1	
Perform when a modal dialog is opened.		2	<no function=""></no>		

• DialogOpen

在作业栏中组态 "Set Value "功能, 链接到内部变量 "Online\_error" (Online\_Fehler). 其值被置为 2。

冬	1-16	
---	------	--

					SCHEDULER
Name		Event		Description	Comment
DialogClose	DialogClose When dialog is closed			Perform when a modal dialog is close	d. 🔨
DialogOpen		When dialog is opened		Perform when a modal dialog is open	ed.
Zykl_1min		1 Minute	2	Perform every minute.	
		<			>
Job			×	++ ==	Function list
Name	Zykl_1min		1	🗆 SetValue	•
Event	1 Minute	-		Tag (Out)	CPU\Online_Fehler
	1			Value	2
Perform every minute		2	<no function=""></no>		

据此可以判断 CPU 是否在线。

"Connection disconnected" 显示时消息窗口打开或关闭,每次值加1。 "Cycl\_1min" 作业把值设为2。

组态一个 I/O 域显示 "CPU offline" 状态,按照下图组态动画可见性。

fling		
Offline (Text Field)		∾ ۹(×
Crag     Online_Fehler	Object state Hidden	Visibility
Type Integer	Range from 0	to 1 🕂
f	fline Offline (Text Field)	Offline (Text Field)         Image: Contract of the state of the

放置上了3个互相重叠的 I/O 域。可以组态在模板中去显示 CPU 状态。

## 2 WinCC flexible 2008 SP1 中的步骤

#### 2.1 介绍

从 WinCC flexible 2008 开始,可以实现"Incoming" 事件。

当系统消息被触发并显示时,会触发 "Incoming" 事件。可以为事件组态函数 或脚本。当触发系统消息时,就执行组态的函数或脚本。

#### 2.2 支持的操作面板

OP 77B, xP 177, xP 277, MP 377, Mobile Panel 277, Mobile Panel 277 IWLAN, Mobile Panel 277F IWLAN, WinCC flexible Runtime

#### 2.3 显示 "System Messages" 编辑器

WinCC flexible 默认不显示 "System messages" 编辑器。通过下列方式激活 "System messages" 编辑器 "Tools > Settings... > Workbench > Settings for project window > Display all entries"。

#### 2.4 配置 CPU 状态

一旦使能系统消息显示,必须为消息分配事件。

在项目窗口中选择 "Messages", 然后打开 "System messages" 编辑器。 有几个选项可以为系统消息实现 "Incoming" 事件。

在此例中,连接到一个圆的外观。当连接断开时,圆为红色,当连接建立 时,圆为绿色。

按下例来显示 CPU 状态。



No.	步骤							
4.	组态一个圆,在	"外观"下组态动画	•	V				
	Circle_1 (Circle)							
	<ul> <li>Properties</li> <li>Animations</li> <li>Appearance</li> <li>Diagonal Movement</li> <li>Horizontal Movement</li> <li>Vertical movement</li> <li>Direct Movement</li> <li>Visibility</li> </ul>	🗹 Enabled	App	Appearance				
		Tag	Yalue 🔺 Foreground Background Color	Flashing				
		Tag_1		No No				
		Integer  Binary  Bit						