

Bausteinbeschreibung

Symbolname	FB_SYS_BASICS		
Titel	System Grundfunktionen		
Familie	System		
Autor	M.Glarner		
Ablauffähig auf	<input type="checkbox"/> S7-300	<input type="checkbox"/> S7-400	<input checked="" type="checkbox"/> S7-1200 <input checked="" type="checkbox"/> S7-1500
Sprache	<input checked="" type="checkbox"/> SCL	<input type="checkbox"/> AWL	<input type="checkbox"/> KOP/FUP
Version	2.40		
Datum	27.11.2018		
Optimiert	<input checked="" type="checkbox"/> ja	<input type="checkbox"/> nein	
Warnungen	-		



Kurzbeschreibung

Folgende Grundfunktionen werden bereitgestellt:

- > Standard Blinktakt und deren positive Flanken
- > Zwei frei definierbare Blinktakte
- > Log.0 und Log. 1 Bit
- > Auswertung CPU Neustart (Richtimpuls)
- > Speicher für Projektdaten
- > Aktuelle, minimale und maximale Zykluszeit berechnen inkl. Resetfunktion
- > Zustandsanzeige LED „MAINT“ z.B. Forceauftrag aktiv
- > Checksumme für Standard- und Safetyprogramm auslesen und überwachen

Eingangsparameter

e_HW_Kennung_CPU	HW_IO	HW-Kennung der CPU
Default Aktualwert: Globale Systemkonstante "Local-Common"		

Durchgangparameter

ea_Ctrl	Struct	UDT_SYS_BASICS_Ctrl
ea_Para	Struct	UDT_SYS_BASICS_Para

UDT_SYS_BASICS_Ctrl

Blinktakt.100ms	BOOL	Blinktakt 100ms (10Hz)
Blinktakt.200ms	BOOL	Blinktakt 200ms (5Hz)
Blinktakt.400ms	BOOL	Blinktakt 400ms (2.5Hz)
Blinktakt.500ms	BOOL	Blinktakt 500ms (2Hz)
Blinktakt.800ms	BOOL	Blinktakt 800ms (1.25Hz)
Blinktakt.1s	BOOL	Blinktakt 1s (1Hz)
Blinktakt.1.6s	BOOL	Blinktakt 1.6s (0.625Hz)
Blinktakt.2s	BOOL	Blinktakt 2s (0.5Hz)
Blinktakt.xFrei1	BOOL	Blinktakt frei definierbar Nr.1 Der Takt kann über den Parameter „ea_Para.Blinktakt.Takt_Frei1“ vorgegeben werden
Blinktakt.xFrei2	BOOL	Blinktakt frei definierbar Nr.2 Der Takt kann über den Parameter „ea_Para.Blinktakt.Takt_Frei2“ vorgegeben werden

FP.Blinktakt.100ms	BOOL	Positive Flanke Blinktakt 100ms (10Hz)
FP.Blinktakt.200ms	BOOL	Positive Flanke Blinktakt 200ms (5Hz)
FP.Blinktakt.400ms	BOOL	Positive Flanke Blinktakt 400ms (2.5Hz)
FP.Blinktakt.500ms	BOOL	Positive Flanke Blinktakt 500ms (2Hz)
FP.Blinktakt.800ms	BOOL	Positive Flanke Blinktakt 800ms (1.25Hz)
FP.Blinktakt.1s	BOOL	Positive Flanke Blinktakt 1s (1Hz)
FP.Blinktakt.1.6s	BOOL	Positive Flanke Blinktakt 1.6s (0.625Hz)
FP.Blinktakt.2s	BOOL	Positive Flanke Blinktakt 2s (0.5Hz)
FP.Blinktakt. xFrei1	BOOL	Positive Flanke Blinktakt frei definierbar Nr.1 Der Takt kann über den Parameter „ea_Para.Blinktakt.Takt_Frei1“ vorgegeben werden
FP.Blinktakt. xFrei2	BOOL	Positive Flanke Blinktakt frei definierbar Nr.2 Der Takt kann über den Parameter „ea_Para.Blinktakt.Takt_Frei2“ vorgegeben werden

CPU_Neustart.FP	BOOL	CPU Neustart: positive Flanke (Richtimpuls) Bit ist nach einem Neustart der CPU für einen Zyklus logisch 1.
CPU_Neustart.Impuls	BOOL	CPU Neustart: Impuls Bit ist nach einem Neustart der CPU für eine gewisse Zeit logisch 1. Die Dauer des Impulses kann im Parameter „ea_Para.TW_Neustart_Impuls“ eingestellt werden.
CPU_Neustart.FN_Impuls	BOOL	CPU Neustart: negative Flanke Impuls Bit ist nach dem Neustartimpuls für einen Zyklus logisch 1.

Log0	BOOL	Bit ist immer logisch 0
Log1	BOOL	Bit ist immer logisch 1

Zykluszeit.act	REAL	Zykluszeit aktuell [ms] Die aktuelle Zykluszeit wird mit dem Befehl „RUNTIME“ berechnet.
Zykluszeit.min	REAL	Zykluszeit minimal [ms] Der Wert wird neu berechnet, nachdem das Bit „ea_Ctrl.Zykluszeit.cmd_reset“ gesetzt wurde.
Zykluszeit.max	REAL	Zykluszeit maximal [ms] Der Wert wird neu berechnet, nachdem das Bit „ea_Ctrl.Zykluszeit.cmd_reset“ gesetzt wurde.
Zykluszeit.cmd_reset	BOOL	Zykluszeit min. und max. reseten (HMI=S;SPS=R) Durch setzen dieses Bits werden die beiden Werte min. und max. Zykluszeit neu berechnet. Anschliessend wird das Bit wieder zurückgesetzt.
Systeminfos_bearbeiten	BOOL	Systeminformationen bearbeiten Damit die Bearbeitungszeit des Bausteins etwas reduziert werden kann, werden die folgenden Systeminformationen nur ausgewertet, wenn dieses Bit log.1 ist: > LED „MAIN“ aktiv > Alle Checksummenwerte
LED_MAINT_on	BOOL	LED "MAINT" aktiv (z.B. durch Forceauftrag)



S7-1500



S7-1200

Checksum.Standard.act	String[23]	Checksumme des Standardprogramm aktuell
Checksum.Standard.save	String[23]	Checksumme des Standardprogramm gespeichert Die aktuelle Checksumme kann mit „ea_Ctrl.CheckSum.Standard.cmd_save“ gespeichert werden. Falls noch keine Checksumme gespeichert wurde, wird der Text , not save' angezeigt.
Checksum.Standard.stat_save_act	BOOL	Status: gespeicherte Checksumme des Standardprogramm aktuell Die aktuelle Checksumme stimmt mit der gespeicherten Checksumme überein. Mit dieser Information kann z.B. eine Meldung auf dem HMI ausgegeben werden, damit der Anwender erkennt, dass eine Programmänderung durchgeführt wurde.
Checksum.Standard.cmd_save	BOOL	Befehl: Checksumme Standardprogramm speichern (HMI=S;SPS=R) Durch setzen dieses Bits wird die aktuelle Checksumme gespeichert. Anschliessend wird das Bit wieder zurückgesetzt.
Checksum.Standard.TimeStamp.last_change	BOOL	Zeitpunkt der letzten Programmänderung im Standardprogramm Wird bei jeder Änderung im Standardprogramm aktualisiert.
Checksum.Standard.TimeStamp.first_change	BOOL	Zeitpunkt der ersten Programmänderung im Standardprogramm nach speichern Wird aktualisiert, sobald die aktuelle und gespeicherte Checksumme nicht mehr übereinstimmen.
Checksum.Standard.TimeStamp.save	BOOL	Zeitpunkt Checksumme Standardprogramm gespeichert Wird aktualisiert, sobald die aktuelle und gespeicherte Checksumme nicht mehr übereinstimmen.
Checksum.Safety.act	String[11]	Checksumme des Safetyprogramm aktuell Bei einer Standard CPU wird der Text ,no Safety“ angezeigt.
Checksum.Safety.save	String[11]	Checksumme des Safetyprogramm gespeichert Die aktuelle Checksumme kann mit „ea_Ctrl.CheckSum.Safety.cmd_save“ gespeichert werden. Falls noch keine Checksumme gespeichert wurde, wird der Text , not save' angezeigt.
Checksum.Safety.stat_save_act	BOOL	Status: gespeicherte Checksumme des Safetyprogramm aktuell Die aktuelle Checksumme stimmt mit der gespeicherten Checksumme überein. Mit dieser Information kann z.B. eine Meldung auf dem HMI ausgegeben werden, damit der Anwender erkennt, dass eine Programmänderung durchgeführt wurde.
Checksum.Safety.cmd_save	BOOL	Befehl: Checksumme Safetyprogramm speichern (HMI=S;SPS=R) Durch setzen dieses Bits wird die aktuelle Checksumme gespeichert. Anschliessend wird das Bit wieder zurückgesetzt.
Checksum.Safety.TimeStamp.last_change	BOOL	Zeitpunkt der letzten Programmänderung im Safetyprogramm Wird bei jeder Änderung im Safetyprogramm aktualisiert.
Checksum.Safety.TimeStamp.first_change	BOOL	Zeitpunkt der ersten Programmänderung im Safetyprogramm nach speichern Wird aktualisiert, sobald die aktuelle und gespeicherte Checksumme nicht mehr übereinstimmen.
Checksum.Safety.TimeStamp.save	BOOL	Zeitpunkt Checksumme Safetyprogramm gespeichert Wird aktualisiert, sobald die aktuelle und gespeicherte Checksumme nicht mehr übereinstimmen.

UDT_SYS_BASICS_Para

Projektdaten.Bezeichnung	String[50]	Projektbezeichnung
Projektdaten.Firma	String[25]	Firma Firma, welche die Software programmiert hat.
Projektdaten.Version	String[10]	Version Version, der letzten Softwareanpassung
Projektdaten.Datum	Date	Datum Datum, der letzten Softwareanpassung
Projektdaten.Autor	String[20]	Autor Autor, der letzten Softwareanpassung
<hr/>		
Blinktakt.Takt_Frei1	TIME	Blinktakt frei definierbar Nr.1 Die Zeitvorgabe entspricht der Periodendauer, somit ist dieser Takt nur für die Hälfte dieser Zeit logisch 1
Blinktakt.Takt_Frei2	TIME	Blinktakt frei definierbar Nr.2 Die Zeitvorgabe entspricht der Periodendauer, somit ist dieser Takt nur für die Hälfte dieser Zeit logisch 1
<hr/>		
TW_Neustart_Impuls	TIME	Timerwert Neustart Impuls

Funktion

Der Baustein sollte am Anfang des OB1 zyklisch aufgerufen werden.
Der Blinktaktmerker in der Gerätekonfiguration wird nicht benötigt.

Versionshistorie

2.40 27.11.2018 M.Glarner

- > Simulierbarkeit mit SIMATIC S7-PLCSIM (Advanced)
- > ENO-Bit immer logisch 1

2.30 14.2.2018 M.Glarner

- > 4x den Schreibfehler „chance“ in „change“ angepasst

2.20 18.1.2018 M.Glarner

- > Bausteineigenschaften optimieren

2.10 14.09.2017 M.Glarner

- > Zusätzliche Informationen können im HMI ausgewählt werden.

2.00 13.09.2017 M.Glarner

- > Checksumme für Standard- und Safetyprogramm auslesen und überwachen
- > Aktuelle, minimale und maximale Zykluszeit mit dem Befehl „RUNTIME“ berechnen
- > Baustein nur noch lauffähig ab Firmware V2.1 (S7-1500) und V4.2 (S7-1200)

1.00 21.08.2016 M.Glarner

- > Erstversion