

# **ABB Drives**

**Applikationshandledning  
Adaptiv blockprogrammering**





Adaptiv blockprogrammering

## **Applikationshandledning**

3AFE64527312 Rev C  
SV  
GÄLLER FRÅN: 08.04.2005



# Innehåll

---

## *Innehåll*

### **Introduktion till handledningen**

Kapitelöversikt .....	7
Kompatibilitet .....	7
Säkerhetsinstruktioner .....	7
Läsare .....	7
Användning .....	7
Kompletterande dokumentation .....	8

### **Adaptiv blockprogrammering**

Kapitelöversikt .....	9
Vad är adaptiv blockprogrammering? .....	9
Hur man bygger ett program .....	10
Anslutning av programmet till drivsystemet .....	11
Hur man styr exekveringen av programmet .....	11

### **Funktionsblock**

Kapitelöversikt .....	13
Generella regler .....	13
Blockingångar .....	13
Parametervärde som heltalsinsignal .....	14
Hur blocket hanterar insignalerna .....	14
Hur man väljer insignal .....	14
Konstant som heltalsinsignal .....	15
Hur man ställer in och ansluter insignalen .....	15
Parametervärde som en boolesk insignal .....	16
Hur blocket hanterar insignalerna .....	16
Hur man väljer insignal .....	16
Konstant som en boolesk insignal .....	17
Hur man ställer in och ansluter insignalen .....	17
Textsträng som insignal .....	17
Hur man väljer insignal .....	17
Funktionsblock .....	18
ABS .....	18
ADD .....	18
OCH .....	18
BITWISE .....	19
JÄM-FÖRARE .....	19
COUNT .....	20
DPOT .....	21

EVENT .....	21
FILTER .....	22
MASK-SET .....	22
MAX .....	22
MIN .....	23
MULDIV .....	23
NEJ .....	23
ELLER .....	24
PI .....	24
PI-BAL .....	25
PI-BIPOLAR .....	25
RAMP .....	25
SR .....	27
SWITCH-B .....	27
SWITCH-I .....	28
TOFF .....	28
TON .....	29
TRIGG .....	29
XELLER .....	30

### ***Driftvärden och parametrar i ACS800 Standardtillämpningsprogram***

Kapitelöversikt .....	31
Driftvärden .....	31
Parametrar .....	32

### ***Schemablanketter för kunder***

Kapitelöversikt .....	39
-----------------------	----

# Introduktion till handledningen

---

## Kapitelöversikt

Kapitlet ger allmän information om handledningen.

## Kompatibilitet

Handledningen överensstämmer med de tillämpningsprogram för frekvensomriktare där adaptiv blockprogrammering ingår.

## Säkerhetsinstruktioner

Följ alla säkerhetsföreskrifter som medföljer frekvensomriktaren.

- Läs igenom säkerhetsinstruktionerna **fullständigt** före installation, driftsättning eller drift av frekvensomriktaren. De fullständiga säkerhetsinstruktionerna återges i början av Hårdvaruhandledningen.
- Läs de **programvaruspecifika varningarna och noterna** innan några grundinställningar för funktioner ändras. För varje funktion ges varningar och noter i "Beskrivning av systemprogramvara", i det avsnitt som beskriver de användarprogrammerbara parametrarna.

## Läsare

Läsaren av denna handledning förväntas:

- vara förtrogen med elinstallationsteknik, elektroniska komponenter och elschemasymboler.
- ha erfarenhet av eller utbildning i installation drift eller service av drivsystem från ABB.

## Användning

Denna handledning är avsedd att användas tillsammans med "Beskrivning av systemprogramvara" för frekvensomriktarens tillämpningsprogram. "Beskrivning av systemprogramvara" innehåller grundläggande information om frekvensomriktarens parametrar inklusive parametrarna för adaptiv blockprogrammering. Denna handledning ger detaljerad information om adaptiv blockprogrammering:

- vad adaptiv blockprogrammering är
- hur man bygger ett program
- hur funktionsblocken fungerar
- hur man dokumenterar ett program

- parametrarna och driftvärdena i ACS800 Standardtillämpningsprogram som är av vikt för adaptiv blockprogrammering.

## **Kompletterande dokumentation**

I användardokumentationen för frekvensomriktaren ingår även:

- "Beskrivning av systemprogramvara" (aktuell handledning levereras med frekvensomriktaren)
- Hårdvaruhandledning (aktuell handledning levereras med frekvensomriktaren)
- Handledningar/tillägg för tillvalsutrustning och program (aktuella handledningar medföljer leveransen).



# Adaptiv blockprogrammering

---

## Kapitelöversikt

Detta kapitel beskriver grunderna för adaptiv blockprogrammering och ger handledning i hur man bygger ett program.

## Vad är adaptiv blockprogrammering?

Normalt styr användaren drivsystemet via parametrar. Varje enskild parameter har en fast uppsättning alternativ eller ett begränsat inställningsområde. Parametrarna gör programmeringen enkel, men begränsar valfriheten. Det går inte att anpassa driften i närmare detalj. Adaptiv blockprogrammering tillåter friare inställning, utan behov av något speciellt programmeringsverktyg eller -språk.

- Programmet bygger på funktionsblock.
- Manöverpanelen fungerar som programmeringsverktyg.
- Användaren kan dokumentera programmet genom att rita det på blockschemablanketter.

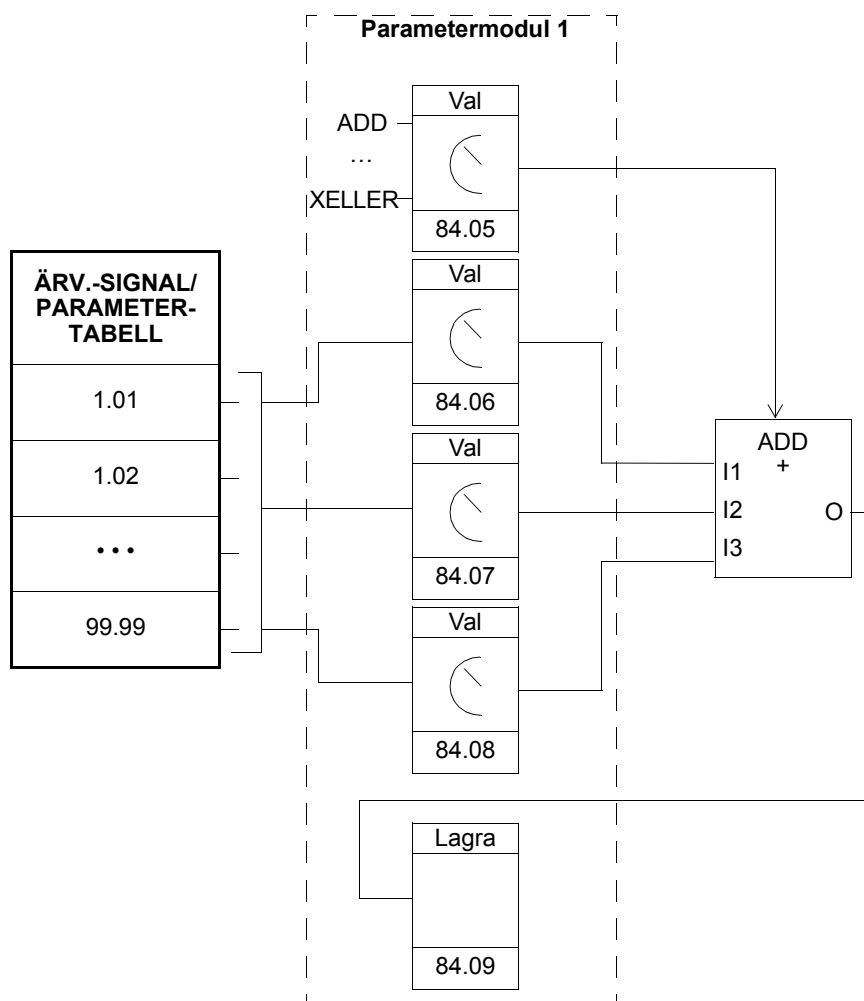
Max storlek för ett adaptivt blockprogram är 15 funktionsblock. Programmet kan bestå av flera separata funktioner.

## Hur man bygger ett program

Programmeraren kopplar samman funktionsblocken med varandra till en parametermodul (Block Parameter Set). Parametermodulerna används även för att läsa värden från och överföra data till frekvensomriktarens tillämpningsprogram. Varje parametermodul består av fem parametrar.

Figuren nedan visar användningen av parametermodul 1 i ACS800 Standardtillämpningsprogram (parametrarna 84.05 till 84.09):

- Parameter 84.05 väljer funktionsblockstyp.
- Parameter 84.06 väljer källan som ingång I1 på funktionsblocket är ansluten till.
- Parameter 84.07 väljer källan som ingång I2 på funktionsblocket är ansluten till.
- Parameter 84.08 väljer källan som ingång I3 på funktionsblocket är ansluten till.
- Parameter 84.09 lagrar värdet på funktionsblockets utgång. Användaren kan inte ändra det parametervärdet.



## Anslutning av programmet till drivsystemet

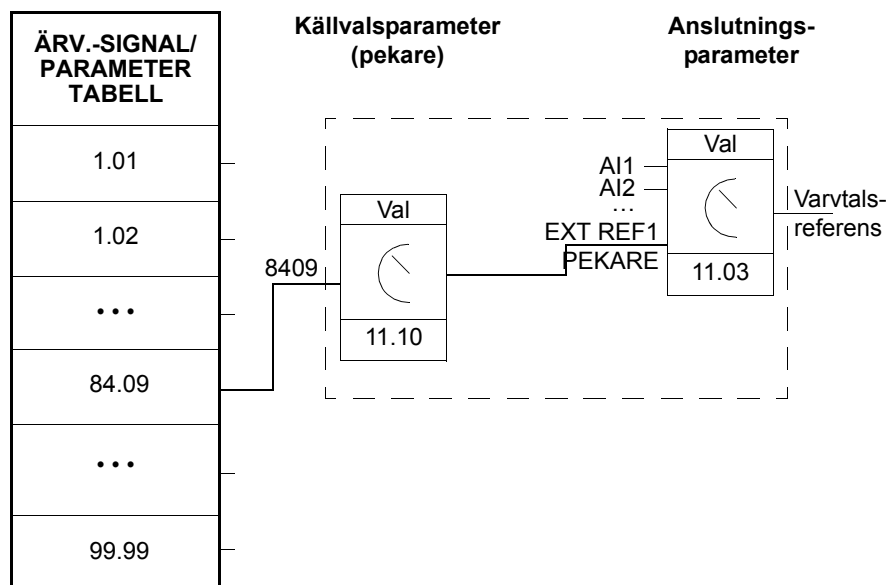
Utdata från det adaptiva blockprogrammet måste kopplas till frekvensomriktarens tillämpningsprogram. För detta ändamål behöver användaren två parametrar:

- en anslutningsparameter och
- en källvalsparameter (pekare).

Figuren nedan visar anslutningsprincipen.

### Exempel:

Utsignalen från det adaptiva blockprogrammet lagras i parameter 84.09. Diagrammet visar hur man kan använda detta värde som varvtalsreferens REF1 i ACS800 Standardtillämpningsprogram.



## Hur man styr exekveringen av programmet

Det adaptiva blockprogrammet exekverar funktionsblocken i nummerordning, samtliga block med samma exekveringstid. Detta kan inte ändras av användaren. Användaren kan:

- välja driftsätt för programmet (stopp, start, ändring)
- välja exekveringstiden för programmet
- radera eller lägga till block.



# Funktionsblock

---

## Kapitelöversikt

Detta kapitel beskriver funktionsblocken.

## Generella regler

**Ingång I1 måste användas (får inte lämnas oansluten).** Användning av övriga ingångar I2, I3, etc. är valfri för de flesta block. Som tumregel gäller att en oansluten ingång inte påverkar utsignalen från blocket.

## Blockingångar

Blocken använder tre ingångsformat:

- heltal
- booleska värden
- textsträngar

Vilket format som används beror på blocket. Till exempel kräver blocket ADD heltal som insignal, medan blocket ELLER kräver booleska insignaler. Endast blocket EVENT kräver insignaler i form av textsträngar.

---

**Viktigt:** Ingångarna till ett block läses av då exekveringen av blocket inleds, dvs. inte samtidigt för alla block.

---

## Parametervärde som heltalsinsignal

### Hur blocket hanterar signalerna

Blocket läser in det valda värdet som ett heltal.

---

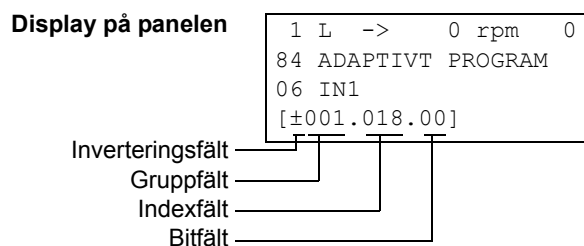
**Viktigt:** Parametern som är vald som insignal ska vara ett reellt tal eller ett heltal. Om värdet inte har heltalsformat konverteras det som standard automatiskt av blocket. Heltalsskalningen (fältbuss) för varje parameter beskrivs i *Beskrivning av systemprogramvara*.

---

### Hur man väljer insignal

- Bläddra till parametern för val av insignal till blocket och övergå till ändringsläge (Enter).
- Ställ in värdena för inverterings-, grupp-, index- och bitfälten i enlighet med adressen från vilken signalen ska läsas (dubbelpil- och piltangenterna).

Figuren nedan visar displayen när valparametern för ingång I1 är i ändringsläge. Insignalvärdet inverteras om det står ett minustecken ( - ) i inverteringsfältet. Bitvalsfältet har ingen inverkan på insignaler som är heltal eller textsträngar.



**Exempel:** Analog ingång AI1 har värdet 5,8 V i en frekvensomriktare utrustad med ACS800 Standardtillämpningsprogram. Hur ansluts den signalen till MAX-blocket i det adaptiva blockprogrammet? Vilket är värdet vid blockingången?

AI1 ansluts till blocket på följande sätt:

- Bläddra till valparametern för ingång I1 och övergå till ändringsläge (Enter).
- Ställ in värdet i gruppfältet till 1 och värdet i indexfältet till 18. (Värdet på AI1 lagras internt som ärvärdessignal 1.18.)

Värdet vid blockets ingång är 5800, eftersom heltalsskalningen av ärvärdessignal 1.18 ger  $0,001 \text{ V} = 1$  (enligt *Beskrivning av systemprogramvara*).

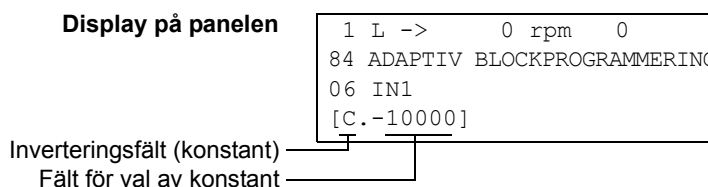
## Konstant som heltalsinsignal

*Hur man ställer in och ansluter insignalen*

### Alternativ 1

- Bläddra till parametern för val av insignal till blocket och övergå till ändringsläge (Enter).
- Välj C i inverteringsfältet (dubbelpil- och piltangenter). Raden får ett nytt utseende. Resten av raden blir nu ett konstantfält.
- Ge konstantfältet ett konstant värde (dubbelpil- och piltangenter).
- Acceptera med Enter.

Figuren nedan visar displayen när valparametern för ingång I1 är i ändringsläge och konstantfältet visas. Konstanten kan ha ett värde från -32768 till 32767. Det konstanta värdet kan inte ändras medan programmet exekveras.



### Alternativ 2

- Ställ in konstanten till en av de parametrar som är reserverade för konstanter.
- Koppla det konstanta värdet till ett block på vanligt sätt, med hjälp av valparametern för ingången.

Konstanterna kan ändras medan det adaptiva blockprogrammet exekveras. De kan anta värden från -8388608 till 8388607.

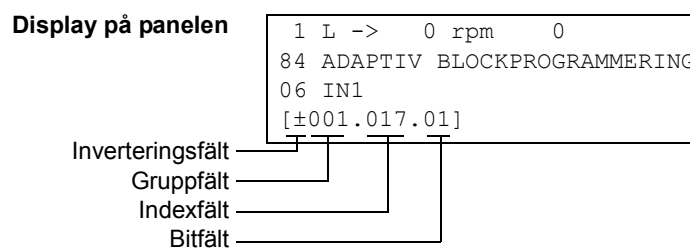
## Parametervärde som en boolesk insignal

### Hur blocket hanterar signalerna

- Blocket läser det valda värdet som ett heltal.
- Blocket använder biten som är definierad av bitfältet som booleska insignal.

Bitvärde 1 motsvarar det booleska värdet sant medan 0 motsvarar det booleska värdet falskt.

**Exempel:** Figuren nedan visar värdet på valparametern för ingång I1 när ingången är ansluten till en bit som indikerar status för digital ingång DI2. (I ACS800 Standardtillämpningsprogram lagras de digitala ingångarnas tillstånd internt som ärvärdessignal 1.17, DI6-1 STATUS. Bit 1 motsvarar DI2, bit 0 motsvarar DI1.)



### Hur man väljer insignal

Se avsnittet [Parametervärde som heltalsinsignal](#) ovan.

---

**Viktigt:** Parametern som är vald som insignal ska ha ett förpackat booleskt värde (binärt dataord). Se "Beskrivning av systemprogramvara".

---



## Konstant som en boolesk insignal

### *Hur man ställer in och ansluter insignalen*

- Bläddra till parametern för val av insignal till blocket och övergå till ändringsläge (Enter).
- Välj C i inverteringsfältet (dubbelpil- och piltangenter). Resten av raden ändras till ett fält för val av konstant.
- Skriv in konstanten. Om det booleska värdet "sant" önskas, sätt konstanten till -1. Om det booleska värdet "falskt" önskas, sätt konstanten till 0.
- Acceptera med Enter.

## Textsträng som insignal

### *Hur man väljer insignal*

Endast blocket EVENT kräver insignaler i form av textsträngar.

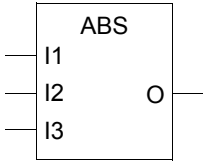
För de val som kan göras med avseende på ingången, se avsnittet [Parametervärde som heltalsinsignal](#) ovan. Bitvals-fältet har ingen inverkan.

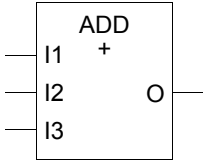
---

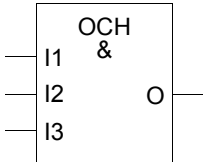
**Viktigt:** Parametern som vald som ingång måste ha ett värde i form av en textsträng. I ACS800 Standardtillämpningsprogram finns det parametrar i grupp 85 ANV KONST som kan användas som textstränginsignaler.

---

## Funktionsblock

<b>ABS</b>	<b>Typ</b>	Aritmetisk funktion
	<b>Blocksymbol</b>	
	<b>Funktion</b>	<p>Utsignalen är lika med absolutvärdet av insignal I1 multiplicerad med I2 och dividerad med I3.</p> $O =  I1  \cdot I2 / I3$
	<b>Anslutningar</b>	<p>Insignal I1, I2 och I3: 24 bitars heltal (23 bitar + tecken)</p> <p>Utsignal (O): 24 bitars heltal (23 bitar + tecken)</p>

<b>ADD</b>	<b>Typ</b>	Aritmetisk funktion
	<b>Blocksymbol</b>	
	<b>Funktion</b>	<p>Utsignalen är lika med summan av insignalerna.</p> $O = I1 + I2 + I3$
	<b>Anslutningar</b>	<p>Insignal I1, I2 och I3: 24 bitars heltal (23 bitar + tecken)</p> <p>Utsignal (O): 24 bitars heltal (23 bitar + tecken)</p>

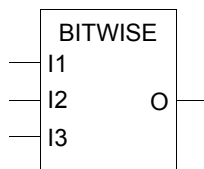
<b>OCH</b>	<b>Typ</b>	Logisk funktion																																													
	<b>Blocksymbol</b>																																														
	<b>Funktion</b>	<p>Utsignalen är sann om samtliga insignaler är sanna. Annars är utsignalen falsk. Sanningstabell:</p> <table border="1" data-bbox="496 1585 1385 1944"> <thead> <tr> <th>I1</th> <th>I2</th> <th>I3</th> <th>O (binärt)</th> <th>O (värde på displayen)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Falsk (alla bitar 0)</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Falsk (alla bitar 0)</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Falsk (alla bitar 0)</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Falsk (alla bitar 0)</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Falsk (alla bitar 0)</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Falsk (alla bitar 0)</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Falsk (alla bitar 0)</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Sann (alla bitar 1)</td> <td>-1</td> </tr> </tbody> </table>	I1	I2	I3	O (binärt)	O (värde på displayen)	0	0	0	Falsk (alla bitar 0)	0	0	0	1	Falsk (alla bitar 0)	0	0	1	0	Falsk (alla bitar 0)	0	0	1	1	Falsk (alla bitar 0)	0	1	0	0	Falsk (alla bitar 0)	0	1	0	1	Falsk (alla bitar 0)	0	1	1	0	Falsk (alla bitar 0)	0	1	1	1	Sann (alla bitar 1)	-1
I1	I2	I3	O (binärt)	O (värde på displayen)																																											
0	0	0	Falsk (alla bitar 0)	0																																											
0	0	1	Falsk (alla bitar 0)	0																																											
0	1	0	Falsk (alla bitar 0)	0																																											
0	1	1	Falsk (alla bitar 0)	0																																											
1	0	0	Falsk (alla bitar 0)	0																																											
1	0	1	Falsk (alla bitar 0)	0																																											
1	1	0	Falsk (alla bitar 0)	0																																											
1	1	1	Sann (alla bitar 1)	-1																																											

**Anslutningar** Insignal I1, I2 och I3: Booleska värden  
 Utsignal (O): 24 bitars heltal (förpackat booleskt värde)

**BITWISE**

**Typ** Aritmetisk funktion

**Blocksymbol**



**Funktion** Blocket jämför bitarna i tre 24 bitars indataord och bildar utgångsbitarna på följande sätt:

$$O = (I1 \text{ OR } I2) \text{ AND } I3$$

**Exempel**

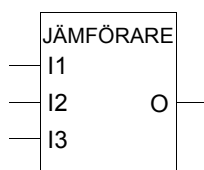
I1	I2	I3	O
0	0	0	0
0	1	0	0
1	0	0	0
1	1	0	0
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	1

**Anslutningar** Insignal I1: 24 bitars heltal (förpackat booleskt värde)  
 Insignal I2: 24 bitars heltal (förpackat booleskt värde)  
 Insignal I3: 24 bitars heltal (förpackat booleskt värde)  
 Utsignal (O): 24 bitars heltal (förpackat booleskt värde)

**JÄM-  
FÖRARE**

**Typ** Jämförande funktion

**Blocksymbol**



**Funktion** Utgångsbitar 0, 1 och 2:  
 - Om I1 > I2, O = ...001 (utgångsbit 0 sätts.)  
 - Om I1 = I2, O = ...010 (utgångsbit 1 sätts.)  
 - Om I1 < I2, O = ...100 (utgångsbit 2 sätts.)  
 Utgångsbit 3  
 - Om I1 > I2, O = ...1xxx (utgångsbit 3 sätts och förblir satt tills I1 < I2 - I3, varefter bit 3 återställs.)  
 Utgångsvärde på displayen:

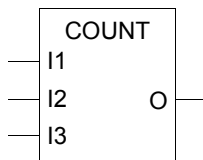
bit 0	bit 1	bit 2	bit 3	O (värde på displayen)
0	0	0	0	0
1	0	0	0	1
0	1	0	0	2
0	0	1	0	4
0	0	0	1	8
1	0	0	1	9
0	1	0	1	10
0	0	1	1	12

**Anslutningar** Insignal I1, I2 och I3: 24 bitars heltal (23 bitar + tecken)  
 Utsignal (O): 24 bitars heltal (förpackat booleskt värde)

**COUNT**

**Typ** Räknarfunktion

**Blocksymbol**

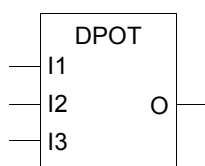


**Funktion** Räknarfunktionen räknar positiva flanker på insignal I1.  
 Räknaren återställs vid positiv flank på insignal I2 och dess värde begränsas till det som sätts på I3.  
 I1: Triggingång  
 I2: Återställning  
 I3: Mazvärde för räknaren (B0...B19 -> 0...1048575)  
 O: Räknarens värde (B0...B19 -> 0...1048575), och räknarens status (B20).  
 B20 = 1: Räknaren har nått sitt maxvärde, eller insignal I3 är negativ.

**Anslutningar** Insignal I1 och I2: Booleska värden  
 Insignal I3: 24 bitars heltal (20 bitar används av räknaren)  
 Utsignal (O): 24 bitars heltal (20 bitar för räknarvärdet och 4 bitar för indikering)

**DPOT****Typ**

Räknarfunktion

**Blocksymbol****Funktion**

Den digitalt styrda rampfunktionen ökar eller minskar utsignalen O beroende på styrsignalerna I1 och I2. Insignal I1 rampar utsignalen i positiv riktning och I2 i negativ riktning. Om båda insignalerna är aktiva händer ingenting. Steget definieras av insignal I3.

Insignal I1: Räkna upp

Insignal I2: Räkna ner

Insignal I3: Rampsteg i positiv/negativ riktning (steg/s).

**Viktigt:** Funktionen beräknar internt med 48 bitars noggrannhet för att undvika offsetfel. .

**Anslutningar**

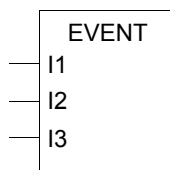
Insignal I1 och I2: Booleska värden

Insignal I3: 24 bitars heltal (23 bitar + tecken)

Utsignal (O): 24 bitars heltal (23 bitar + tecken)

**EVENT****Typ**

Händelsefunktion

**Blocksymbol****Funktion**

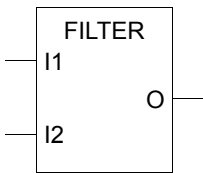
Insignal I1 triggas händelsen. I2 väljer det parameterindex från vilket händelsemeddelandet (textsträngen) läses. I3 väljer typen av händelse (varning eller fel).

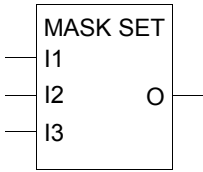
I1	I2	I3	Orsak
0->1			blocket aktiverar händelsen
0			blocket deaktiverar händelsen
	I2		innehåll i händelsemeddelandet
		0	typ av händelse: varning
		1	typ av händelse: fel
		2	typ av händelse: händelse

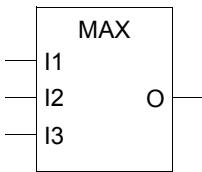
**Anslutningar**

Insignal I1, I3: 24 bitars heltal (23 bitar + tecken)

Insignal I2: Sträng (obligatorisk)

<b>FILTER</b>	<b>Typ</b>	Filtereringsfunktion
	<b>Blocksymbol</b>	
	<b>Funktion</b>	<p>Utsignalen är det filtrerade värdet av insignal I1. Insignal I2 är filtertiden.</p> $O = I1 \cdot (1 - e^{-t/I2})$ <p><b>Viktigt:</b> Funktionen arbetar internt med 48 bitars noggrannhet för att undvika offsetfel.</p>
	<b>Anslutningar</b>	<p>Insignal I1: 24 bitars heltal (23 bitar + tecken)</p> <p>Insignal I2: 24 bitars heltal (23 bitar + tecken). Ett motsvarar 1 ms.</p> <p>Utsignal (O): 24 bitars heltal (23 bitar + tecken)</p>

<b>MASK-SET</b>	<b>Typ</b>	Logisk funktion																																																
	<b>Blocksymbol</b>																																																	
	<b>Funktion</b>	<p>Blockfunktionen sätter eller återställer I1 beroende på värdet på I2.</p> <p>Insignal I1: Ordinsignal</p> <p>Insignal I2: Sätt-ordinsignal</p> <p>Insignal I3: Sätt/återställ I2 till I1.</p>																																																
		<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th colspan="4">Exempel med SET</th> </tr> <tr> <th>I1</th> <th>I2</th> <th>I3</th> <th>O</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Sann</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>Sann</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>Sann</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>Sann</td> <td>1</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th colspan="4">Exempel med RESET</th> </tr> <tr> <th>I1</th> <th>I2</th> <th>I3</th> <th>O</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Falsk</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>Falsk</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>Falsk</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>Falsk</td> <td>0</td> </tr> </tbody> </table>	Exempel med SET				I1	I2	I3	O	0	0	Sann	0	1	0	Sann	1	1	1	Sann	1	0	1	Sann	1	Exempel med RESET				I1	I2	I3	O	0	0	Falsk	0	1	0	Falsk	1	1	1	Falsk	0	0	1	Falsk	0
Exempel med SET																																																		
I1	I2	I3	O																																															
0	0	Sann	0																																															
1	0	Sann	1																																															
1	1	Sann	1																																															
0	1	Sann	1																																															
Exempel med RESET																																																		
I1	I2	I3	O																																															
0	0	Falsk	0																																															
1	0	Falsk	1																																															
1	1	Falsk	0																																															
0	1	Falsk	0																																															
	<b>Anslutningar</b>	<p>Insignal I1: 24 bitars heltal (förpackat booleskt värde)</p> <p>Insignal I2: 24 bitars heltal (förpackat booleskt värde)</p> <p>Insignal I3: Booleskt</p> <p>Utsignal (O): 24 bitars heltal (förpackat booleskt värde)</p>																																																

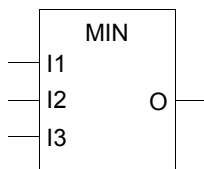
<b>MAX</b>	<b>Typ</b>	Jämförande funktion
	<b>Blocksymbol</b>	

**Funktion** Utsignalen är lika med den högsta av insignalerna.  
 $O = \text{MAX}(I1, I2, I3)$

**Anslutningar** Insignal I1, I2 och I3: 24 bitars heltal (23 bitar + tecken)  
 Utsignal (O): 24 bitars heltal (23 bitar + tecken)

**MIN** **Typ** Jämförande funktion

**Blocksymbol**

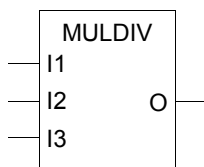


**Funktion** Utsignalen är lika med den lägsta av insignalerna.  
 $O = \text{MIN}(I1, I2, I3)$

**Anslutningar** Insignal I1, I2 och I3: 24 bitars heltal (23 bitar + tecken)  
 Utsignal (O): 24 bitars heltal (23 bitar + tecken)

**MULDIV** **Typ** Aritmetisk funktion

**Blocksymbol**

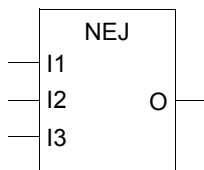


**Funktion** Utsignalen är lika med produkten av insignal I1 och insignal I2, dividerad med insignal I3.  
 $O = (I1 \cdot I2) / I3$

**Anslutningar** Insignal I1, I2 och I3: 24 bitars heltal (23 bitar + tecken)  
 Utsignal (O): 24 bitars heltal (23 bitar + tecken)

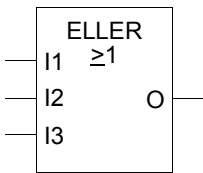
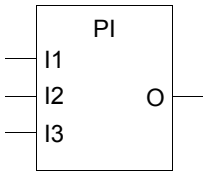
**NEJ** **Typ** -

**Blocksymbol**



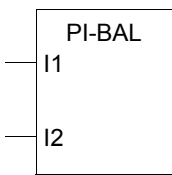
**Funktion** Detta block saknar funktion.

**Anslutningar** -

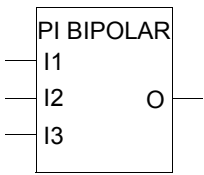
<b>ELLER</b>	<b>Typ</b>	Logisk funktion																																								
	<b>Blocksymbol</b>																																									
	<b>Funktion</b>	<p>Utsignalen är sann om minst en av insignalerna är sann. Sanningstabell:</p> <table border="1"> <thead> <tr> <th>I1</th> <th>I2</th> <th>I3</th> <th>O (binärt)</th> <th>O (värde på displayen)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Falsk (alla bitar 0)</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Sann (alla bitar 1)</td> <td>-1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Sann (alla bitar 1)</td> <td>-1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Sann (alla bitar 1)</td> <td>-1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Sann (alla bitar 1)</td> <td>-1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Sann (alla bitar 1)</td> <td>-1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Sann (alla bitar 1)</td> <td>-1</td> </tr> </tbody> </table>	I1	I2	I3	O (binärt)	O (värde på displayen)	0	0	0	Falsk (alla bitar 0)	0	0	0	1	Sann (alla bitar 1)	-1	0	1	0	Sann (alla bitar 1)	-1	0	1	1	Sann (alla bitar 1)	-1	1	0	0	Sann (alla bitar 1)	-1	1	1	0	Sann (alla bitar 1)	-1	1	1	1	Sann (alla bitar 1)	-1
I1	I2	I3	O (binärt)	O (värde på displayen)																																						
0	0	0	Falsk (alla bitar 0)	0																																						
0	0	1	Sann (alla bitar 1)	-1																																						
0	1	0	Sann (alla bitar 1)	-1																																						
0	1	1	Sann (alla bitar 1)	-1																																						
1	0	0	Sann (alla bitar 1)	-1																																						
1	1	0	Sann (alla bitar 1)	-1																																						
1	1	1	Sann (alla bitar 1)	-1																																						
	<b>Anslutningar</b>	<p>Insignal I1, I2 och I3: Booleska värden          Utsignal (O): 24 bitars heltal (förpackat booleskt värde)</p>																																								
<b>PI</b>	<b>Typ</b>	PI-regulator																																								
	<b>Blocksymbol</b>																																									
	<b>Funktion</b>	<p>Utsignalen är lika med insignal I1 dividerad med I2/100 plus den heltalskonverterade signalen I1 multiplicerad med I3/100.  <math display="block">O = I1 \cdot I2/100 + (I3/100) \cdot \int I1</math> <b>Viktigt:</b> Funktionen beräknar internt med 48 bitars noggrannhet för att undvika offsetfel.</p>																																								
	<b>Anslutningar</b>	<p>Insignal I1: 24 bitars heltal (23 bitar + tecken)          Insignal I2:          - 24 bitars heltal (23 bitar + tecken)          - Förstärkningsfaktor. 100 motsvarar 1. 10 000 motsvarar 100.          Insignal I3:          - Integrerande koefficient. 100 motsvarar 1. 10 000 motsvarar 100.          Utsignal (O): 24 bitars heltal (23 bitar + tecken). Området begränsas till 0...10000.</p>																																								



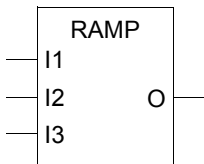
---

<b>PI-BAL</b>	<b>Typ</b>	Initieringsblock för PID-regulatorn
	<b>Blocksymbol</b>	
	<b>Funktion</b>	<p>Detta block initierar först PI-blocket. När signalen på ingång I1 blir sann skriver blocket värdet på I2 till utgången från PI-blocket. När I1 blir falsk släpper blocket utgången från PI-regulatorblocket, vilket fortsätter med normal drift från den satta utgången.</p> <p><b>Viktigt:</b> Detta block får endast användas tillsammans med PI-blocket. Blocket måste följa efter PI-blocket.</p>
	<b>Anslutningar</b>	<p>Insignal I1: Booleskt värde          Insignal I2: 24 bitars heltal (23 bitar + tecken)</p>

---

<b>PI-BIPOLAR</b>	<b>Typ</b>	PI-regulator
	<b>Blocksymbol</b>	
	<b>Funktion</b>	<p>Se blocket PI.          Undantag          Utsignalområde (O): -10000...10000.</p>

---

<b>RAMP</b>	<b>Typ</b>	Rampfunktion
	<b>Blocksymbol</b>	

---

**Funktion**

Blocket använder insignal I1 som referensvärde. Stegvärdena (insignalerna I2 och I3) ökar eller minskar utsignalen O så länge utsignalen avviker från gränsvärdet I1. När  $O = I1$  blir utsignalen stabil.

Insignal I1: Referensvärde

Insignal I2: Steg i positiv riktning (steg/s). Öka utsignalen när  $O < I1$ .

Insignal I3: Steg i negativ riktning (steg/s). Minska utsignalen när  $O > I1$ .

$$O_n = O_{n-1} + I2 \text{ när } I1 > O$$

$$O_n = O_{n-1} - I3 \text{ när } I1 < O$$

$$O_n = I1 \text{ när } I1 = O$$

**Exempel:**

Insignal I1: 0 -> 150 -> -100 -> 0

Insignal I2: 100 steg/s

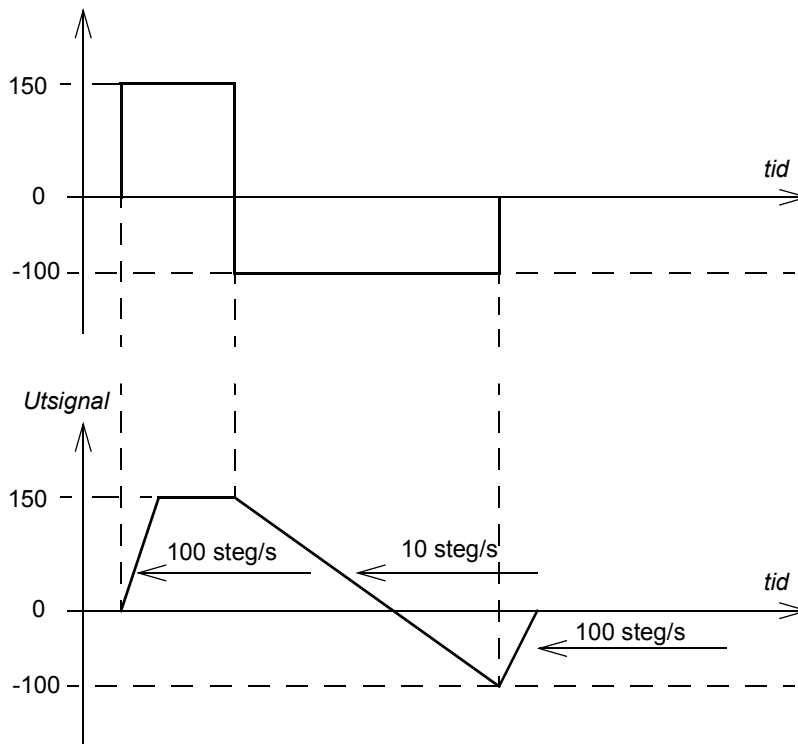
Insignal I3: 10 steg/s

Utsignal:

Riktning uppåt: Rampsteg från insignal I2

Going down: Rampsteg från insignal I3

*Insignal I1*

**Anslutningar**

Insignal I1, I2 och I3: 24 bitars heltal (23 bitar + tecken)

Utsignal (O): 24 bitars heltal (23 bitar + tecken)

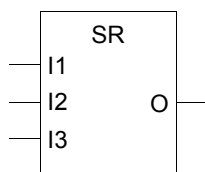
**Viktigt:** Funktionen beräknar internt med 48 bitars noggrannhet för att undvika offsetfel.

**SR**

Typ

Logisk funktion

Blocksymbol



Funktion

Sättnings-/återställningsblock. Insignal I1 sätter utgången, medan insignalerna I2 och I3 återställer den.

- Om I1, I2 och I3 är falska förändras inte utsignalen.
- Om I1 är sann och I2 och I3 är falska blir utsignalen sann.
- Om I2 eller I3 är sanna blir utsignalen falsk.

I1	I2	I3	O (binärt)	O (värde på displayen)
0	0	0	Ut	Ut
0	0	1	Falsk (alla bitar 0)	0
0	1	0	Falsk (alla bitar 0)	0
0	1	1	Falsk (alla bitar 0)	0
1	0	0	Sann (alla bitar 1)	-1
1	0	1	Falsk (alla bitar 0)	0
1	1	0	Falsk (alla bitar 0)	0
1	1	1	Falsk (alla bitar 0)	0

Anslutningar

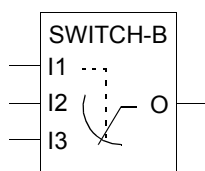
Insignal I1, I2 och I3: Booleska värden  
 Utsignal (O): 24 bitars heltal (23 bitar + tecken)

**SWITCH-B**

Typ

Logisk funktion

Blocksymbol



Funktion

Utsignalen är lika med insignal I2 om insignal I1 är sann, respektive lika med insignal I3 om insignal I1 är falsk.

I1	I2	I3	O	O (värde på displayen)
0	I2	I3	I3	Sann = -1
1	I2	I3	I2	Falsk = 0

Anslutningar

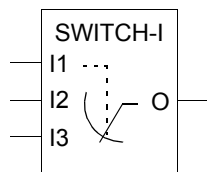
Insignal I1, I2 och I3: Booleska värden  
 Utsignal (O): 24 bitars heltal (förpackat booleskt värde)

**SWITCH-I**

Typ

Logisk funktion

Blocksymbol



Funktion

Utsignalen är lika med insignal I2 om insignal I1 är sann, respektive lika med insignal I3 om insignal I1 är falsk.

I1	I2	I3	O
0	I2	I3	I3
1	I2	I3	I2

Anslutningar

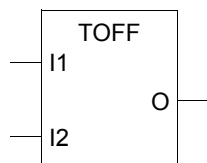
Insignal I1: Booleskt värde  
 Insignal I2 och I3: 24 bitars heltal (23 bitar + tecken)  
 Utsignal (O): 24 bitars heltal (23 bitar + tecken)

**TOFF**

Typ

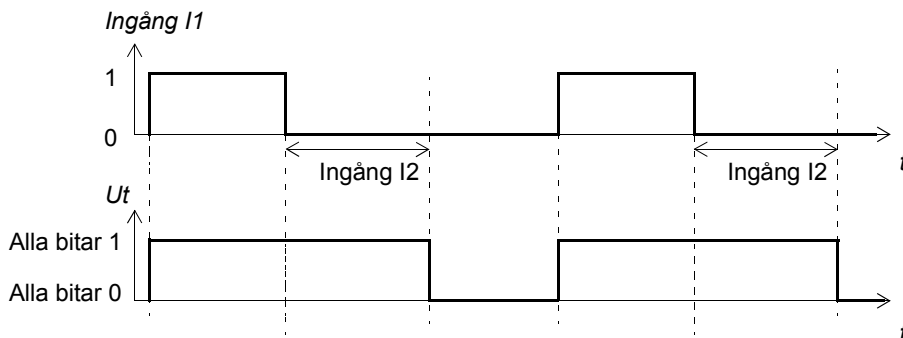
Tidkrets

Blocksymbol



Funktion

Utsignalen är sann om insignal I1 är sann. Utsignalen är falsk om insignal I1 har varit falsk under en tid lika med eller längre än I2.



Värden på displayen: Sann = -1, falsk = 0.

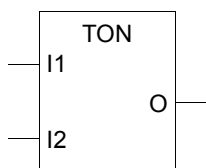
Anslutningar

Insignal I1: Booleskt värde  
 Insignal I2: 24 bitars heltal (23 bitar + tecken). Ett motsvarar 1 ms.  
 Utsignal (O):  
 - 24 bitars heltal (förpackat booleskt värde)

**TON**

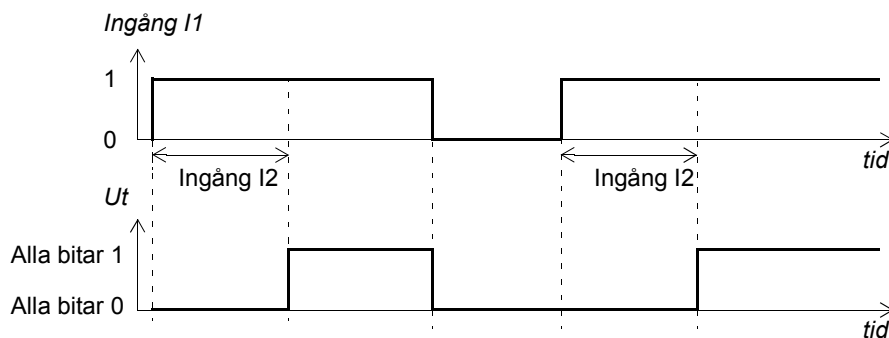
Typ Tidskrets

**Blocksymbol**



**Funktion**

Utsignalen är sann om insignal I1 har varit sann under en tid lika med eller längre än I2. Utsignalen är falsk om insignalen är falsk.



Värden på displayen: Sann = -1, falsk = 0.

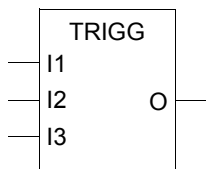
**Anslutningar**

Insignal I1: Booleskt värde  
 Insignal I2: 24 bitars heltal (23 bitar + tecken). 1 motsvarar 1 ms.  
 Utsignal (O): 24 bitars heltal (förpackat booleskt värde)

**TRIGG**

Typ Tidskrets

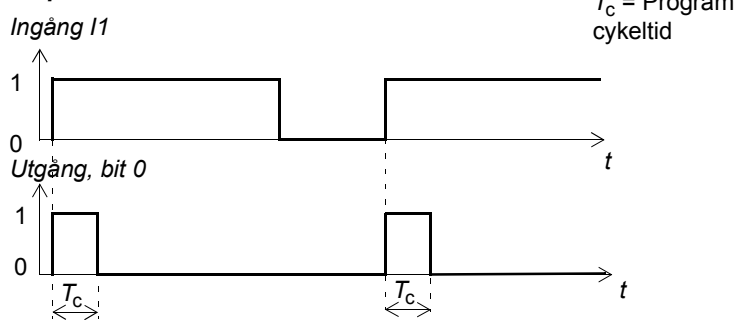
**Blocksymbol**



**Funktion**

Positiv flank hos insignal I1 sätter utgångsbit 0 under en programcykel.  
 Positiv flank hos insignal I2 sätter utgångsbit 1 under en programcykel.  
 Positiv flank hos insignal I3 sätter utgångsbit 2 under en programcykel.

**Exempel:**

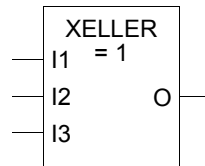


**Anslutningar** Insignal I1, I2 och I3: Booleska värden  
 Utsignal (O):  
 - 24 bitars heltal (23 bitar + tecken)

**XELLER**

**Typ** Logisk funktion

**Blocksymbol**



**Funktion** Utsignalen är sann om insignalen är sann. Annars är utsignalen falsk.  
 Sanningstabell:

I1	I2	I3	O (binärt)	O (värde på displayen)
0	0	0	Falsk (alla bitar 0)	0
0	0	1	Sann (alla bitar 1)	-1
0	1	0	Sann (alla bitar 1)	-1
0	1	1	Falsk (alla bitar 0)	0
1	0	0	Sann (alla bitar 1)	-1
1	0	1	Falsk (alla bitar 0)	0
1	1	0	Falsk (alla bitar 0)	0
1	1	1	Sann (alla bitar 1)	-1

**Anslutningar** Insignal I1, I2 och I3: Booleska värden  
 Utsignal (O):  
 - 24 bitars heltal (23 bitar + tecken)

# Driftvärden och parametrar i ACS800 Standardtillämpningsprogram

## Kapitelöversikt

Detta kapitel listar driftvärden, parametrar och parametervärden för ACS800 Tillämpningsprogram som är av betydelse för adaptiv blockprogrammering.

## Driftvärden

Tabellen nedan listar driftvärden som är av betydelse för det adaptiva blockprogrammet. Förkortningen FbEkv står för fältbussekvivalent.

Index	Signalnamn/Värde	Beskrivning	FbEkv.
09	DRIFTVÄRDEN	Signaler för det adaptiva blockprogrammet	
09.01	AI1 SKALAT VÄRDE	Värde på analog ingång AI1, skalat till ett heltal.	20000 = 10 V
09.02	AI2 SKALAT VÄRDE	Värde på analog ingång AI2, skalat till ett heltal.	20000 = 20 mA
09.03	AI3 SKALAT VÄRDE	Värde på analog ingång AI3, skalat till ett heltal.	20000 = 20 mA
09.04	AI5 SKALAT VÄRDE	Värde på analog ingång AI5, skalat till ett heltal.	20000 = 20 mA
09.05	AI6 SKALAT VÄRDE	Värde på analog ingång AI6, skalat till ett heltal.	20000 = 20 mA
09.06	MASTERSTYRORD	Styrord för huvudreferensdataset, mottaget från masterstationen via fältbussgränssnittet.	-32768 ... 32767
09.07	MASTER REF1	Referens 1 (REF1) i huvudreferensdataset, mottaget från masterstationen via fältbussgränssnittet.	-32768 ... 32767
09.08	MASTER REF2	Referens 2 (REF2) i huvudreferensdataset, mottaget från masterstationen via fältbussgränssnittet.	-32768 ... 32767
09.09	UTÖKAT DS ORD1	Referens 3 (REF3) i hjälpreferensdataset, mottaget från masterstationen via fältbussgränssnittet.	-32768 ... 32767
09.10	UTÖKAT DS ORD2	Referens 4 (REF4) i hjälpreferensdataset, mottaget från masterstationen via fältbussgränssnittet.	-32768 ... 32767
09.11	UTÖKAT DS ORD3	Referens 5 (REF5) i hjälpreferensdataset, mottaget från masterstationen via fältbussgränssnittet.	-32768 ... 32767

## Parametrar

Tabellen nedan listar parametrar och parametervärden som är av betydelse för adaptiv blockprogrammering. Förkortningen FbEkv står för fältbussekvivalent.

Index	Parameternamn / värde	Beskrivning	FbEkv
10	START/STOPP/ROTR	Parametrar via vilka det adaptiva blockprogrammet kan styra start, stopp och rotationsriktning för drivsystemet.	
10.01	EXT1 STRT/STP/ROT		
	PARAM 10.04	Källa vald av <a href="#">10.04</a> .	17
10.02	EXT2 STRT/STP/ROT		
	PARAM 10.05	Källa vald av <a href="#">10.05</a> .	17
10.04	EXT 1 STRT PEKARE	Väljer källan för parameter <a href="#">10.01</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde: - Parameterpekare: Inverterings-, grupp-, index- och bifält. Bitnumret gäller endast för block som hanterar booleska ingångar. - Konstantvärde: Inverterings- och konstantfält. Inverteringsfältet måste ha värdet C för att tillåta inställning av konstanten.	
10.05	EXT 2 STRT PEKARE	Väljer källan för <a href="#">10.02</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
11	VAL AV REFERENS	Parametrar via vilka det adaptiva blockprogrammet kan styra referensen för drivsystemet.	
11.02	VAL EXT1/EXT2		
	PARAM 11.09	Källa vald av <a href="#">11.09</a> .	16
11.03	EXT REF1		
	PARAM 11.10	Källa vald av <a href="#">11.10</a> .	37
11.06	EXT REF2		
	PARAM 11.11	Källa vald av <a href="#">11.11</a> .	38
11.09	EXT 1/2 PEK VAL	Väljer källan för <a href="#">11.02</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
11.10	EXT 1REF PEKARE	Väljer källan för <a href="#">11.03</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
11.11	EXT 2REF PEKARE	Väljer källan för <a href="#">11.06</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
14	RELÄUTGÅNGAR	Parametrar via vilka det adaptiva blockprogrammet kan styra reläutgångarna från frekvensomriktaren.	
14.01	RELÄ RO1 UTGÅNG		
	PARAM 14.16	Källa vald av parameter <a href="#">14.16</a> .	36
14.02	RELÄ RO2 UTGÅNG		



Index	Parameternamn / värde	Beskrivning	FbEkv
	PARAM 14.17	Källa vald av parameter <a href="#">14.17</a> .	36
14.03	RELÄ RO3 UTGÅNG		
	PARAM 14.18	Källa vald av parameter <a href="#">14.18</a> .	36
14.10	DIO STATUS1 RO1		
	PARAM 14.19	Källa vald av parameter <a href="#">14.19</a> .	7
14.11	DIO STATUS1 RO2		
	PARAM 14.20	Källa vald av parameter <a href="#">14.20</a> .	7
14.12	DIO STATUS2 RO1		
	PARAM 14.21	Källa vald av parameter <a href="#">14.21</a> .	7
14.13	DIO STATUS2 RO2		
	PARAM 14.22	Källa vald av parameter <a href="#">14.22</a> .	7
14.14	DIO STATUS3 RO1		
	PARAM 14.23	Källa vald av parameter <a href="#">14.23</a> .	7
14.15	DIO STATUS3 RO2		
	PARAM 14.24	Källa vald av parameter <a href="#">14.24</a> .	7
14.16	RO PEKARE 1	Väljer källan för parameter <a href="#">14.01</a> .	
	-255.255.31 ... +255.255.31/C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
14.17	RO PEKARE 2	Väljer källan för parameter <a href="#">14.02</a> .	
	-255.255.31 ... +255.255.31/C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
14.18	RO PEKARE 3	Väljer källan för parameter <a href="#">14.03</a> .	
	-255.255.31 ... +255.255.31/C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
14.19	RO PEKARE 4	Väljer källan för parameter <a href="#">14.10</a> .	
	-255.255.31 ... +255.255.31/C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
14.20	RO PEKARE 5	Väljer källan för parameter <a href="#">14.11</a> .	
	-255.255.31 ... +255.255.31/C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
14.21	RO PEKARE 6	Väljer källan för parameter <a href="#">14.12</a> .	
	-255.255.31 ... +255.255.31/C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
14.22	RO PEKARE 7	Väljer källan för parameter <a href="#">14.13</a> .	
	-255.255.31 ... +255.255.31/C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
14.23	RO PEKARE 8	Väljer källan för parameter <a href="#">14.14</a> .	

Index	Parameternamn / värde	Beskrivning	FbEkv
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04.</a>	
14.24	RO PEKARE 9	Väljer källan för parameter <a href="#">14.15.</a>	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04.</a>	
15	ANALOGA UTGÅNGAR	Parametrar via vilka det adaptiva blockprogrammet kan styra frekvensomriktarens analoga standardutgångar.	
15.01	ANALOG UTGÅNG 1		
	PARAM 15.11	Källa vald av <a href="#">15.11</a>	17
15.06	ANALOG UTGÅNG 2		
	PARAM 15.12	Källa vald av <a href="#">15.12</a>	16
15.11	AO1 PEKARE	Väljer källan för parameter <a href="#">15.01.</a>	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04.</a>	
15.12	AO2 PEKARE	Väljer källan för parameter <a href="#">15.06.</a>	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04.</a>	
16	SYSTEMSTYRNING	Parametrar via vilka det adaptiva blockprogrammet kan styra frekvensomriktarens systemstyrningångar.	
16.01	DRIFTFRIGIVNING		
	PARAM 16.08	Källa vald av parameter <a href="#">16.08.</a>	15
16.08	PEKARE DRIFTFRIG	Väljer källan för parameter <a href="#">16.01</a>	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04.</a>	
20	GRÄNSER	Parametrar via vilka det adaptiva blockprogrammet kan styra drivsystemets driftsmässiga gränser.	
20.13	MIN MOMENT VAL	Väljer mingränsen för vridmoment	
	PARAM 20.18	Gränsvärde som ges av <a href="#">20.18</a>	20
20.14	MAX MOMENT VAL	Väljer maxgränsen för vridmoment	
	PARAM 20.19	Gränsvärde som ges av <a href="#">20.19</a>	19
20.18	MIN MOMENT PEKARE	Väljer källan för <a href="#">20.13</a>	100 = 1%
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04.</a>	
20.19	MAX MOMENT PEKARE	Väljer källan för <a href="#">20.14</a>	100 = 1%
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04.</a>	

Index	Parameternamn / värde	Beskrivning	FbEkv
22	ACCEL/RETARD	Parametrar via vilka det adaptiva blockprogrammet kan styra acceleration och retardation av drivsystemet.	
22.01	ACC/DEC 1/2 VAL		
	PAR 22.08&09	Accelerations- och retardationstider som ges av parametrarna 22.08 och 22.09	15
22.08	ACC PEKARE	Väljer källan för 22.01	100 = 1 s
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
22.09	DEC PEKARE	Väljer källan för 22.01	100 = 1 s
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
26	MOTOR STYRNING	Parametrar via vilka det adaptiva blockprogrammet kan styra statorns flöde.	
26.06	FLÖDES REF PEKARE	Väljer källan för flödesreferens.	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
40	PID REGULATOR	Parametrar via vilka det adaptiva blockprogrammet kan påverka PID-regleringen av processen.	
40.07	ÄRV1 INGÅNG		
	PARAM 40.25	Källa vald av parameter <a href="#">40.25</a> .	6
40.25	ÄRV1 PEKARE	Väljer källan för <a href="#">40.07</a>	100 = 1%
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
83	ADAPTIV PROG KTRL	Exekveringsstyrning för det adaptiva blockprogrammet.	
83.01	ADAPTIV PROG KTRL	Väljer driftsätt för det adaptiva blockprogrammet.	
	STOPP	Stopp av program. Programmet kan inte ändras.	
	START	Start av program. Programmet kan inte ändras.	
	ÄNDRINGSLÄGE	Stopp för ändring. Programmet kan ändras.	
83.02	ÄNDRA	Väljer kommando för block placerat på den plats som definieras av parameter <a href="#">83.03</a> . Programmet måste vara i ändringsläge. (Se parameter <a href="#">83.01</a> .)	
	NEJ	Startvärde. Värdet återgår automatiskt till NEJ efter att ett ändringskommando har utförts.	

Index	Parameternamn / värde	Beskrivning	FbEkv																											
	FLYTTA UPP	<p>Flyttar blocket på den plats som definieras av parameter <a href="#">83.03</a> samt efterföljande block en plats uppåt. Ett nytt block kan placeras i det tomma utrymmet genom att man programmerar en blockparametermodul på vanligt sätt.</p> <p><b>Exempel:</b> Ett nytt block ska skjutas in mellan aktuellt block nr. 4 (parametrar 84.20 ... 84.25) och nr. 5 (parametrar 84.25 ... 84.29).</p> <p>För att göra detta:</p> <ul style="list-style-type: none"> <li>- Övergå till ändringsläge via parameter <a href="#">83.01</a>.</li> <li>- Välj position nr. 5 som önskad destination för det nya blocket via parameter <a href="#">83.03</a>.</li> <li>- Förskjut blocket i position nr. 5 och alla efterföljande block en position framåt via parameter <a href="#">83.02</a> (val FLYTTA UPP).</li> <li>- Programmera den tömda position nr. 5 via parametrarna 84.25 till 84.29 på vanligt sätt.</li> </ul>																												
	RADERA	Raderar blocket i den position som definieras av parameter <a href="#">83.03</a> och förskjuter alla efterföljande block en position nedåt.																												
83.03	ÄNDRA BLOCK	Definierar blockpositionsnummer för kommandot som valts med parameter <a href="#">83.02</a> .																												
	1 ... 15	Blockpositionsnummer																												
83.04	VAL AV EXEKV TID	Väljer exekveringstid för det adaptiva blockprogrammet. Denna inställning gäller för samtliga block.																												
	12 ms	12 millisekunder																												
	100 ms	100 millisekunder																												
	1000 ms	1000 millisekunder																												
84	ADAPTIVT PROGRAM	Konstruktion och diagnos av det adaptiva blockprogrammet.																												
84.01	STATUS	<p>Visar värdet för det adaptiva blockprogrammets statusord. Tabellen nedan visar de olika bittillstånden och motsvarande värden på panelens display.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Display</th> <th>Innebörd</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>Stoppad</td> </tr> <tr> <td>1</td> <td>2</td> <td>Drift</td> </tr> <tr> <td>2</td> <td>4</td> <td>Fel</td> </tr> <tr> <td>3</td> <td>8</td> <td>Ändring</td> </tr> <tr> <td>4</td> <td>10</td> <td>Kontroll</td> </tr> <tr> <td>5</td> <td>20</td> <td>Pushing</td> </tr> <tr> <td>6</td> <td>40</td> <td>Popping</td> </tr> <tr> <td>8</td> <td>100</td> <td>Initierar</td> </tr> </tbody> </table>	Bit	Display	Innebörd	0	1	Stoppad	1	2	Drift	2	4	Fel	3	8	Ändring	4	10	Kontroll	5	20	Pushing	6	40	Popping	8	100	Initierar	
Bit	Display	Innebörd																												
0	1	Stoppad																												
1	2	Drift																												
2	4	Fel																												
3	8	Ändring																												
4	10	Kontroll																												
5	20	Pushing																												
6	40	Popping																												
8	100	Initierar																												
84.02	FEL PAR	Pekar på den felbehäftade parametern i det adaptiva blockprogrammet.																												
84.05	BLOCK1	Väljer funktionsblock för parametermodul 1.																												
	ABS	Se kapitlet <a href="#">Funktionsblock</a> .																												
	ADD	Se kapitlet <a href="#">Funktionsblock</a> .																												
	OCH	Se kapitlet <a href="#">Funktionsblock</a> .																												
	JÄMFÖRARE	Se kapitlet <a href="#">Funktionsblock</a> .																												
	EVENT	Se kapitlet <a href="#">Funktionsblock</a> .																												
	FILTER	Se kapitlet <a href="#">Funktionsblock</a> .																												
	MAX	Se kapitlet <a href="#">Funktionsblock</a> .																												
	MIN	Se kapitlet <a href="#">Funktionsblock</a> .																												

Index	Parameternamn / värde	Beskrivning	FbEkv
	MULDIV	Se kapitlet <a href="#">Funktionsblock</a> .	
	NEJ	Se kapitlet <a href="#">Funktionsblock</a> .	
	ELLER	Se kapitlet <a href="#">Funktionsblock</a> .	
	PI	Se kapitlet <a href="#">Funktionsblock</a> .	
	PI-BAL	Se kapitlet <a href="#">Funktionsblock</a> .	
	SR	Se kapitlet <a href="#">Funktionsblock</a> .	
	SWITCH-B	Se kapitlet <a href="#">Funktionsblock</a> .	
	SWITCH-I	Se kapitlet <a href="#">Funktionsblock</a> .	
	TOFF	Se kapitlet <a href="#">Funktionsblock</a> .	
	TON	Se kapitlet <a href="#">Funktionsblock</a> .	
	TRIGG	Se kapitlet <a href="#">Funktionsblock</a> .	
	XELLER	Se kapitlet <a href="#">Funktionsblock</a> .	
84.06	IN1	Väljer källan för ingång I1 på parametermodul 1 (BPS1).	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde: - Parameterpekare: Inverterings-, grupp-, index- och bitfält. Bitnumret gäller endast för block som hanterar booleska ingångar. - Konstantvärde: Inverterings- och konstantfält. Inverteringsfältet måste ha ett värde C för att tillåta inställning av konstanten. <b>Exempel:</b> Tillståndet för digital ingång DI2 kopplas till Ingång 1 på följande sätt: - Sätt källvalsparametern (84.06) till +.01.17.01. (Tillämpningsprogrammet lagrar tillståndet för den digitala ingången DI2 i form av bit 1 i ärvärdessignal 01.17.) - Invertera värdet genom att byta tecken på pekarvärdet (-01.17.01.).	
84.07	IN2	Se parameter <a href="#">84.06</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Se parameter <a href="#">84.06</a> .	
84.08	IN3	Se parameter <a href="#">84.06</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Se parameter <a href="#">84.06</a> .	
84.09	UT	Lagrar och visar utsignalen från parametermodul 1.	
...	...	Lagrar och visar utsignalen från parametermodul 15.	
84.79	UT	Lagrar utsignalen från parametermodul 15. Se parameter <a href="#">84.09</a> .	
85	ANV KONST	Lagring av det adaptiva blockprogrammets konstanter och meddelanden.	
85.01	KONST1	Ställer in en konstant för det adaptiva blockprogrammet.	
	-8388608 till 8388607	Heltal.	
85.02	KONST2	Ställer in en konstant för det adaptiva blockprogrammet.	
	-8388608 till 8388607	Heltal.	
85.03	KONST3	Ställer in en konstant för det adaptiva blockprogrammet.	
	-8388608 till 8388607	Heltal.	
85.04	KONST4	Ställer in en konstant för det adaptiva blockprogrammet.	
	-8388608 till 8388607	Heltal.	

Index	Parameternamn / värde	Beskrivning	FbEkv
85.05	KONST5	Ställer in en konstant för det adaptiva blockprogrammet.	
	-8388608 till 8388607	Heltal.	
85.06	KONST6	Ställer in en konstant för det adaptiva blockprogrammet.	
	-8388608 till 8388607	Heltal.	
85.07	KONST7	Ställer in en konstant för det adaptiva blockprogrammet.	
	-8388608 till 8388607	Heltal.	
85.08	KONST8	Ställer in en konstant för det adaptiva blockprogrammet.	
	-8388608 till 8388607	Heltal.	
85.09	KONST9	Ställer in en konstant för det adaptiva blockprogrammet.	
	-8388608 till 8388607	Heltal.	
85.10	KONST10	Ställer in en konstant för det adaptiva blockprogrammet.	
	-8388608 till 8388607	Heltal.	
85.11	STRÄNG1	Lagrar ett meddelande som ska användas i det adaptiva blockprogrammet (blocket EVENT).	
	MEDDEL 1	Meddelande	
85.12	STRÄNG2	Lagrar ett meddelande som ska användas i det adaptiva blockprogrammet (blocket EVENT).	
	MEDDEL 2	Meddelande	
85.13	STRÄNG3	Lagrar ett meddelande som ska användas i det adaptiva blockprogrammet (blocket EVENT).	
	MEDDEL 3	Meddelande	
85.14	STRÄNG4	Lagrar ett meddelande som ska användas i det adaptiva blockprogrammet (blocket EVENT).	
	MEDDEL 4	Meddelande	
85.15	STRÄNG5	Lagrar ett meddelande som ska användas i det adaptiva blockprogrammet (blocket EVENT).	
	MEDDEL 5	Meddelande	
96	EXT AO	Parametrar via vilka det adaptiva blockprogrammet kan styra de analoga tillvalsutgångarna från frekvensomriktaren.	
96.01	EXT AO1		
	PARAM 96.11	Källa vald av parameter <a href="#">96.11</a> .	16
96.06	EXT AO2		
	PARAM 96.12	Källa vald av parameter <a href="#">96.11</a> .	16
96.11	EXT AO1 PEKARE	Väljer källan för <a href="#">96.01</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	
96.12	EXT AO2 PEKARE	Väljer källan för <a href="#">96.06</a> .	
	-255.255.31 ... +255.255.31 / C.-32768 ... C.32767	Parameterpekare eller konstant värde. Se parameter <a href="#">10.04</a> .	

# Schemablanketter för kunder

---

## Kapitelöversikt

Detta kapitel innehåller tre tomma schemablanketter på vilka adaptiva blockprogram kan dokumenteras.













---

**ABB Motors and Machines**

Huvudkontor

S-72170 Västerås

SVERIGE

Telefon +46-21-342000

Telefax +46-21-187841

Internet [www.abb.se](http://www.abb.se)

3AFE64527312 Rev C / SV  
GÄLLER FRÅN: 08.04.2005