GATU- OCH FASTIGHETS-
KONTORET

Projekteringsanvisningar för nya PRIBUSS
PRIPTS
ELC-3 V3.21

av
Tor Thomassen

1996-12-17

Tittel: Projekteringsanvisningar för PRIPTS
Författare: Tor Thomassen
Datum: 96-12-17
Fil: \ny-pri\elc\projanv
Version: 1.0

Projekteringsanvisningar för PRIPTS. ELC-3 V3.21

Detta dokument beskriver projekteringsmöjligheterna för PRIPTS-logiken som finns i ELC-3 version 3.21. PRIPTS-logiken är baserad på PRIBUSS.

PRIPTS-logiken som finns i ELC-3 ägs av Stockholms Gatu- och Fastighetskontor. PRIPTS-logiken får ej användas utan skriftlig medgivande från Stockholms Gatu- och Fastighetskontor.

Detta dokument får inte kopieras utan skriftligt medgivande från Stockholms Gatu- och Fastighetskontor.

Dokumentet beskriver hur man projekterar PRIBUSS-funktionerna i en samordnad trafiksignal som innehåller PRIPTS-logiken. Med PRIBUSS menas den standard som Signalbyrån på Stockholms Gatu- och Fastighetskontor respektive Trafiktjänsten, Stockholm Entreprenad, har använt vid prioritering av kollektivtrafik i trafiksignalanläggningar. För en mer detaljerad analys av PRIPTS-logiken hänvisas *till Peek Traffic User's Description for PRIPTS* eller Gatu- och Fastighetskontorets *Programmeringsanvisningar för PRIPTS* och för en mer detaljerad analys av PRIBUSS-funktionerna hänvisas till Stockholms Gatukontors *PRIBUSS, Projekterings- och Programmeringsanvisningar*.

Referenser:

User's Description PRIPTS in ELC-3.	Peek Traffic
Principiellt programmeringsunderlag för PRIBUSS-funktioner i styrapparatens grundprogram.	Stockholm Konsult
PRIBUSS Projekterings- och programmeringsanvisningar.	Stockholms Gatukontor

Copyright © Stockholms Gatu- och Fastighetskontor

Sammanfattning.

Den nya PRIPTS-logiken bygger på PRIBUSS-funktionerna. PRIPTS-logiken är implementerad i styrapparatens grundprogram, till skillnad från PRIBUSS som var programmerad i styrapparatens controllblock.

PRIPTS-logiken är i första hand tänkt att ersätta PRIBUSS-funktionerna i samordnade system. Alla huvudfunktioner som tidigare fanns i PRIBUSS finns med i den nya PRIPTS-logiken med undantag för *ÖVERGÅNG SAMO TILL OBEROENDE* och *ÅTERSYNK TILL SAMO.*

I nya PRIPTS-logiken finns det uppgraderade versioner av; BF, ÅTS, EF, DEF, AK och DAK. Funkionen *BEVAKA* finns inte kvar, men kan ersättas av en PRIPTS-logik som bara har i uppdrag att prioriteringsbegränsa andra funktioner. Funktionerna *Status* och *Prioriterings-begränsning* finns kvar, men har fått en helt annan utformning. Funktionerna *Plus steg* och *Viss steg* finns inte med. Om man i vissa speciella fall vill använda dessa funktioner får man utnyttja controllblocksprogrammering. Den gamla flyttfunktionen har ersatts med de nya funktionerna FLYTT 1 och FLYTT 2.

PRIBUSS består av flera olika funktioner som t.ex. BF och EF. Om en BF-funktion skulle nyttjas fyllde projektören i en skräddarsydd blankett för BF och programmeraren programmerade flera olika controllblock som var specialgjorda för funktionen BF. Det fanns alltså specialgjorda blanketter och controllblock för varje enskild funktion.

I den nya PRIPTS-logiken finns det ingen särskild BF-funktion eller EF-funktion. Alla typer av funktioner kan programmeras i logiken. I stället för att ha flera olika projekterings- och programmeringsblanketter beroende på vilken funktion som skall användas kommer det i fortsättningen bara att användas en projekteringsblankett.

Copyright © Stockholms Gatu- och Fastighetskontor

Tittel: Projekteringsanvisningar för PRIPTS
Författare: Tor Thomassen
Datum: 96-12-17
Fil: \ny-pri\elc\projanv
Version: 1.0

Innehåll

Sida

2 ---------- Sammanfattning.
4 ---------- Förkortningar och förklaringar.
6 ---------- Exempel Signalplan och signalväxlingsshema.
7 ---------- Beskrivning.
10 ---------- Generella parametrar.
Nr. 1
Nr. 3
11 ---------- Räknare.
Nr. 6 - 9
Nr. 11 - 13
14 ---------- Prioriteringsbegränsning.
Nr. 17-20
16 ---------- Status.
Nr. 26-30
18 ---------- Förlängning och avslut av prioritering.
Nr. 41
Nr. 43
Nr. 48
Nr. 49
21 ---------- Avkorta signalgrupper och ge start till signalgrupper.
Nr. 31
Nr. 33 - 36
Nr. 38 - 41
26 ---------- Luckor och sista-steg.
Nr. 16
Nr. 47
30 ---------- Flytta starter och förhindra starter.
Nr. 21-25
34 ---------- Exempel Beskrivning av generella parametrar och räknare.
35 ---------- Exempel BF1.
36 ---------- Exempel BF2.
37 ---------- Exempel ÅTS.
38 ---------- Exempel EF1.
39 ---------- Exempel EF2.
40 ---------- Exempel DEF.
41 ---------- Exempel DAK.
42 ---------- Exempel AK.

Bilaga ------- Projekteringsblankett.

Copyright © Stockholms Gatu- och Fastighetskontor

Förkortningar och förklaringar.

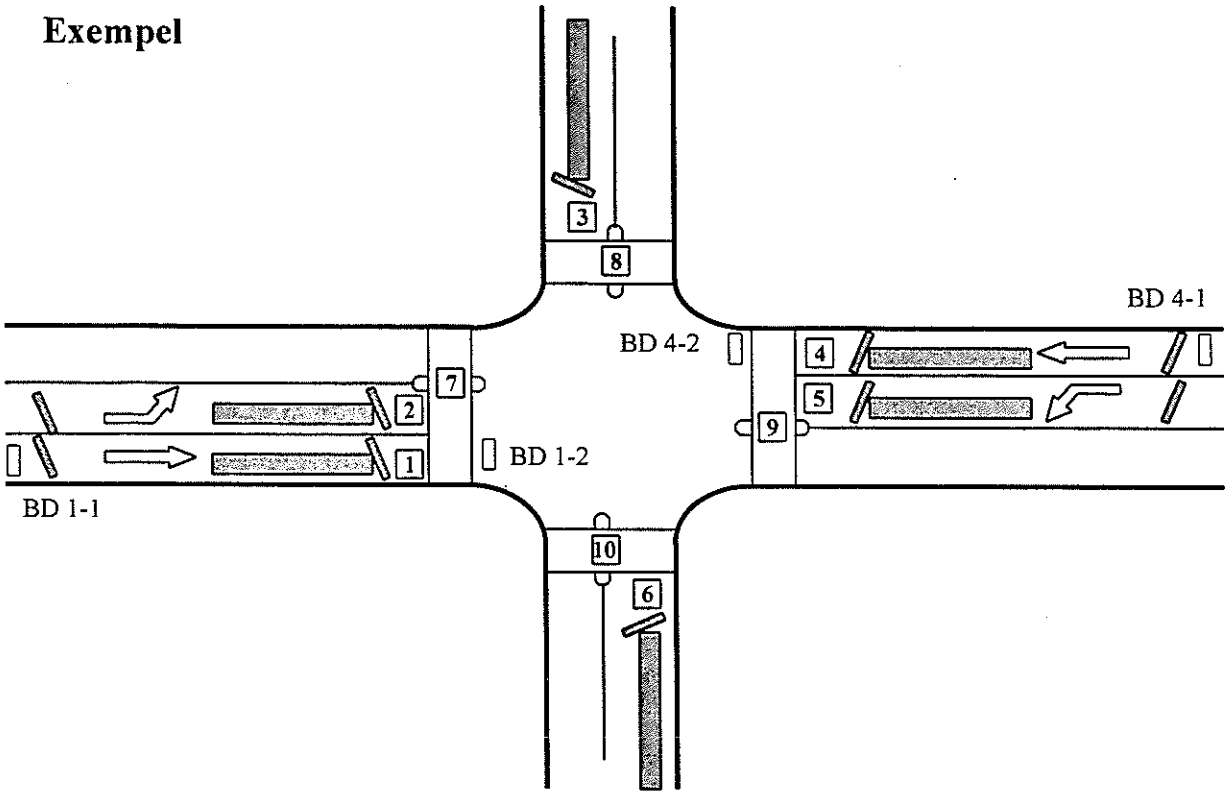
PRIPTS	PRIorization of Public Transport in Signal systems
PRIBUSS	PRIoritering av BUssar i Samordnade Signalsystem
Nr.	Parameter nummer.
ÅTS	ÅterTagenStart.
BF	BussFörlägning.
EF	Extra Fas.
AK	AvKortning.
DAK	DubbelAvKortning.
DEF	Dubbelavkortning med Extra Fas.
Vx1	Växling 1. Första växling vid DAK och DEF d.v.s. avkortning och start till grupper.
Vx2	Växling 2. Andra växling vid DAK och DEF eller avkortnig och start till grupper vid vanlig AK, EF och ÅTS.
För	Förlägning vid BF, ÅTS, EF och DEF.
TO	TimeOut.
AD	Anmälande Detektor.
AVD	AvAnmälande Detektor.
Stp	Förhindra starter.

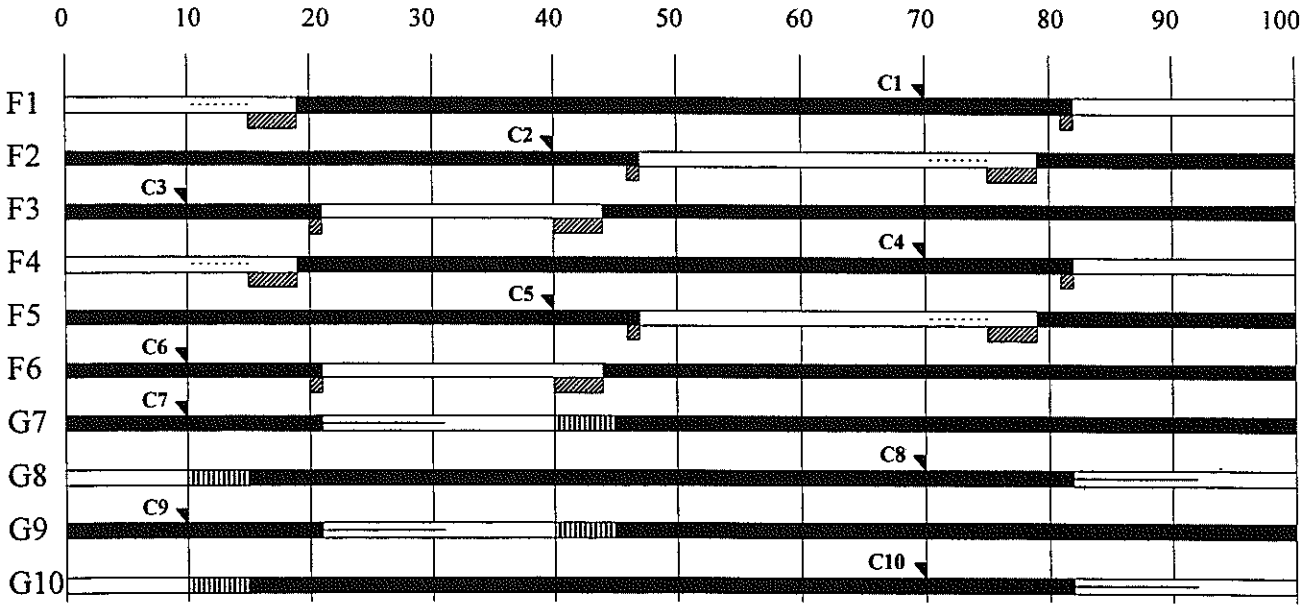
Copyright © Stockholms Gatu- och Fastighetskontor

Lucka	Fönster i omloppet som bestämmer om funktionen skall tillåtas.
Lucka NyBuss	Fönster i omloppet som bestämmer om ny buss skall tillåtas.
Lucka Ny Mot	Fönster i omloppet som bestämmer om ny buss i motriktat körfält skall tillåtas.
Sista-Steg	Sekundsteg i omloppet där funktionen obenhörligen avslutas oberoende av om buss har avanmält sig eller inte.

Copyright © Stockholms Gatu- och Fastighetskontor

Exempel





Start	Signalgrupp	Sekundsteg	Start	Signalgrupp	Sekundsteg
C1	F1	70	C6	F6	10
C2	F2	40	C7	G7	10
C3	F3	10	C8	G8	70
C4	F4	70	C9	G9	10
C5	F5	40	C10	G10	70

Time-Out F1: 12
Time-Out F4: 17

Copyright © Stockholms Gatu- och Fastighetskontor

Beskrivning.

I PRIPTS-logiken för varje styrapparat finns möjlighet att utnyttja 40 olika prioriterings-funktioner (logiker). Varje logik (funktion) är indelad i 5 moduler, nämligen:

1. Start:	Kontrollerar Lucka, Status, Prioriteringsbegränsningar och yttre villkor.
2. Växling 1:	Kontrollerar grupper, avslutar grupper och startar grupper.
3. Växling 2:	Kontrollerar grupper, avslutar grupper och startar grupper.
4. Förlängning:	Förlänger grupper.
5. Avslut:	Avslutar grupper och startar grupper.

1. **Start**modulen används för att kontrollera att alla förutsättningar är uppfyllda innan logiken påbörjar avkortning eller förlängning av signalgrupper. Startmodulen startas när en anmälande buss detekteras. Om luckan inte är aktiv avslutas logiken direkt, men om luckan är aktiv fortsätter startmodulen med att kontrollera status och prioriterings-begränsning. Startmodulen väntar tills alla villkor är uppfyllda under förutsättning att luckan är öppen eller att bussen inte avanmälar sig. När alla villkor är uppfyllda prioriteringsbegränsas alla andra logiker och logiken fortsätter med att avkorta eller förlänga de på förhand definierade signalgrupperna. Om luckan stängs eller om bussen avanmäler sig innan villkoren är uppfyllda avslutas logiken.

2. **Växling 1** används för att ställa grupper till rött samt ge ny start till grupper vid DAK eller DEF. Innan Växling 1 påbörjar sin avkortning av grupper måste vissa angivna grupper vara gröna. Det är lämpligt att kontrollera att de grupper som sedan skall avkortas verkligen är gröna.

3. **Växling 2** används som andra växling under DAK och DEF, men används också när bara en växling skall utföras (t.ex. AK och EF). Innan Växling 2 påbörjar sin avkortning av grupper måste vissa angivna grupper vara gröna.

4. **Förlängning** används för att förlänga bussgruppen och allierade grupper under BF, ÅTS, EF och DEF. Det finns två typer av förlängning; maxtidsförlängning och fråntidsförlängning.

5. **Avslut**modulen aktiveras av avanmälande detektor, Sista-Steg eller när TO=0. Grupper som förlängs kan då sättas till rött och efterföljande grupper kan ges start.

Copyright © Stockholms Gatu- och Fastighetskontor

Om man exempelvis vill programmera en BF, används endast modulerna **Start, Förlängning** och **Avslut**. (Vid samordnad styrning behöver man normalt ej ge ny start till efterföljande signalgrupper eftersom ordinarie start inte förhindras.) Till dessa 5 moduler tillkommer specialfunktioner för räknaren, flytta starter och prioriterings-begränsning.

Räknare

Varje logik har en räknare. Räknaren aktiveras av anmälande detektor och räknaren räknas ner av avanmälande detektor. När Time-Out blivit noll och om tillfarten fortfarande är grön nollställs även räknaren. För att spara plats i minnet finns det möjlighet att använda andra logikers räknare. T.ex. att låta räknaren till en logik i tillfarten sköta all in- och uträkning.

Flytta starter.

Funktionen för flyttning av starter bygger på att om prioriteringen avslutas i en lucka, som defineras mellan vissa angivna sekundsteg så skall efterföljande angivna starter flyttas ett antal sekunder. Det finns möjlighet att ange två olika luckor för att aktivera denna funktion. Luckorna kallas **Flytt1** och **Flytt2**. Funktionerna fungerar på följande sätt: Om prioriteten avslutas innan luckan för Flytt1 blir aktivt flyttas inga starter alls. Om logiken avslutas när luckan för Flytt1 är aktiv flyttas de på förhand definierade starterna ett visst angivet antal sekunder och vidare om logiken avslutas när Flytt1 har blivit passiv och luckan för Flytt2 har blivit aktiv tar Flytt2 över och flyttar de på förhand definierade starterna ett angivet antal sekunder. Självklart så behöver man inte alltid använda ovannämda flyttfunktioner.

Prioriteringsbegränsning.

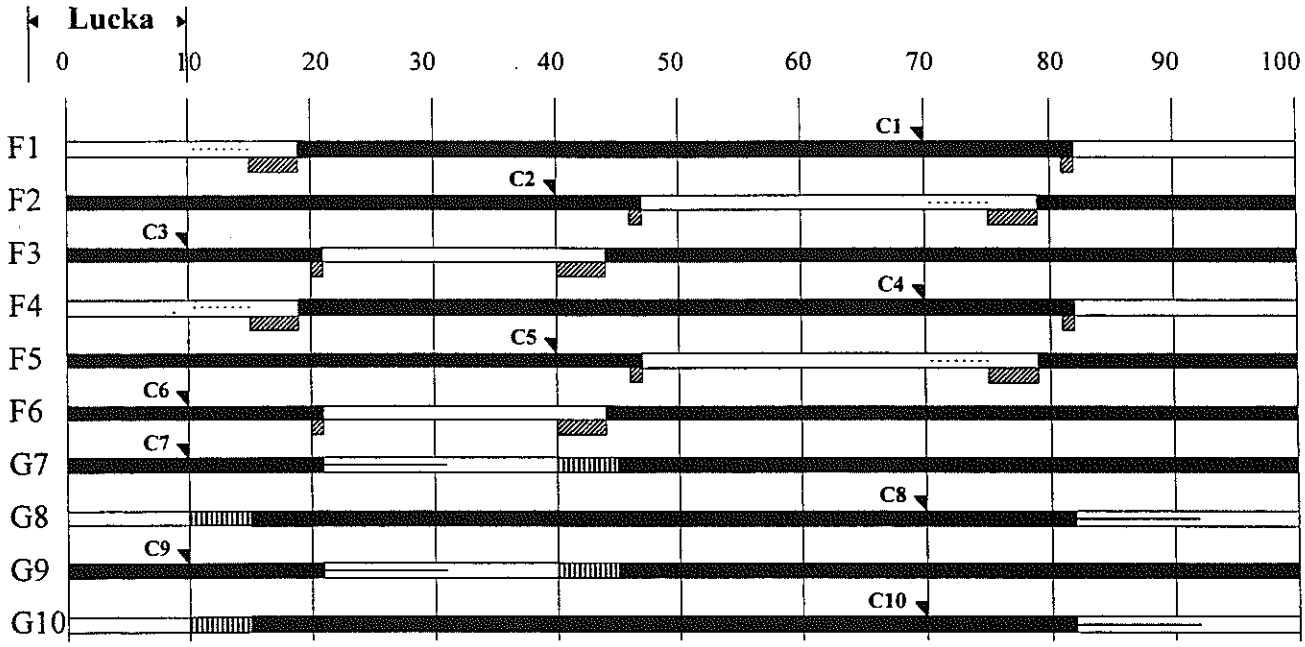
När en viss logik pågår finns det möjlighet att prioriteringsbegränsa alla andra logiker med en individuell tid i sekunder. Alla logiker är prioriteringsbegränsade när någon annan logik är aktiv. Det är normalt bara en logik som kan vara aktiv. Det finns dock möjlighet att välja bort denna begränsning när t.ex. är två korsningar är programmerade i en och samma apparat.

Luckor.

Lucka är den plats i omloppet där en funktion, t.ex. BF, tillåts starta. I PRIBUSS fanns det bara en lucka. I nya PRIPTS-logiken tillkommer det ytterligare två luckor. Lucka beskriver den del av omloppet som en viss funktion tillåts starta. Lucka NyBuss beskriver den del av omloppet där ny buss tillåts förlänga logiken ytterligare. Lucka NyBussMotriktad beskriver den del av omloppet där en ny buss i allierad tillfart tillåts förlänga logiken. Skillnaden mellan PRIBUSS och den nya PRIPTS-logiken är att under EF och ÅTS kunde inte ny buss få förnyad TimeOut på grund av att status inte stämde. I nya PRIPTS-logiken är det enbart luckan NyBuss som bestämmer om TimeOut skall uppdateras när det kommer ny buss under pågående logik.

Copyright © Stockholms Gatu- och Fastighetskontor

I PRIBUSS definierades luckan från sekundsteg xx till sekundsteg yy. Luckan var fast programmerad och om man ville ändra starter i samordningen var man tvungen att ändra luckans öppningstider. I den nya PRIPTS-logiken är luckan definierad i relation till starterna. Om man skall definiera luckan för en BF i t.ex. tillfart F1 är det mera logisk att välja starten till F1 som utgångspunkt därför att vi vet att ett antal sekunder efter starten kommer F1 att bli grön och vi vet också att efter starten till en fientlig grupp kommer tillfarten växla från grönt. Om man t.ex. vill att luckan skall öppnas 13 sekunder efter starten till F1 och stängas 5 sekunder efter starten till F3, och om samordningen sedan behöver en justering och starten till F1 flyttas 5 sekunder åt ena eller andra hållet flyttas luckan med och prioriteringen fungerar som förut. Luckan blir dynamisk.



Lucka BF i tillfart F1: C1+28 - C3

Det är inte tillåtet att öppna eller stänga luckor på sekundsteg 0. Det är heller inte tillåtet att flytta starter till sekundsteg 0 eller placera sista-steg på sekundsteg 0.

Copyright © Stockholms Gatu- och Fastighetskontor

Generella parametrar.

1	Typ av funktion	BF,ÅTS..... 1,2..	
2			
3	Aktiv i tidplan	Px 1/27	

Nr. 1 Typ av funktion.

Här anges typ av funktion. Denna parameter fungerar endast som information till programmerare och projektör.

1	Typ av funktion	BF,ÅTS..... 1,2..	**BF**

Nr. 3 Aktiv i tidplan.

Här anges de tidplaner där logiken är tillåten.

3	Aktiv i tidplan	Px 1/27	**P2, P3, P4**

Copyright © Stockholms Gatu- och Fastighetskontor

Räknare.

6	Förlängande detektor	SGx Mi Fi sek.	DP	
7	Räknare	Rx	PR	
8	Anmälande detektor	Det x	In	
9	Avanmälande detektor	Det x	In	
10				
11	Time-Out	x sek.	sek	
12	Tillfart	SGx	SG	
13	Motsatt räknare	Rx	PR	

<u>**Nr. 6** Förlängande detektor.</u>

Här definieras den detektorlogik som skall förlänga busstillfarten vid prioritering. Varje logik aktiverar en utgång som används till att förlänga den prioriterade tillfarten. Utgången knyts till en detektorlogik och projektören har då möjlighet att påverka logikens utformning. I praktiken används maxtidsförlängning och/eller fråntidsförlängning med ett intervall på 0.1 sekund. Utgången passiviseras vid avanmälan eller sista-steg.

6	Förlängande detektor	SGx Mi Fi sek.	DP	**SG1 M 0.1 F 0.1 sek.**

Detektorlogiken skall förlänga F1 med ett maxtidsintervall och ett fråntidsintervall på 0.1 sekunder.

<u>**Nr. 7** Räknare.</u>

Här anges nummer på räknaren som skall påverka logiken. Alla logiker har var sin räknare. Om den egna logikens räknare skall användas fyller man i logikens nummer i blanketten. Om en annan logiks räknare skall användas anges denna logikens nummer.

7	Räknare	Rx	PR	**R1**

För att spara plats i minnet programmerar man normalt bara en räknare per tillfart och låter sedan även andra logiker i tillfarten titta på denna räknare.

Ex. I tillfart F1 finns BF, ÅTS, EF och AK. I BF-logiken programmeras räknaren och man låter de andra logikerna använda samma räknare. I ÅTS, EF och AK anges då bara nummret på räknaren för BF-logiken i parameter Nr. 7 och parameter 8-12 lämnas tomma.

Copyright © Stockholms Gatu- och Fastighetskontor

Nr. 8 Anmälande detektor.

Här anges anmälande detektoringång.

8	Anmälande detektor	Det x	In	**BD 1-1**

Används annan räknare än logikens egen räknare (denna räknare anges i Nr. 7) behöver man ej programmera denna parameter, d.v.s. man anger då endast vilken logik som är räknare i parameter Nr. 7.

Nr. 9 Avanmälande detektor.

Här anges avanmälande detektoringång.

9	Avanmälande detektor	Det x	In	**BD 1-2**

Används annan räknare än logikens egen räknare (denna räknare anges i Nr. 7) behöver man ej programmera denna parameter, d.v.s. man anger då endast vilken logik som är räknare i parameter Nr. 7.

Nr. 11 Time-Out.

Här anges tillfartens TimeOut. Time-Out är den körtid (intervall) som varje buss tilldelas. Skillnaden mellan t.ex. maxtidsintervallet för en detektor och Time-Out är att TO räknas oberoende av vilken färg tillfarten har. TO uppdateras varje gång det kommer en buss om Lucka NyBuss är aktiv. I tidigare version av PRIBUSS var man ofta tvungen att använda en längre Time-Out i EF än t.ex. BF därför att växlingstiden kunde vara längre än den Time-Out som fanns i BF. I nya PRIPTS-logiken finns det möjlighet att uppgradera Time-Out när gruppen blir grön (Nr. 46). Detta gör det möjligt att använda samma Time-Out för alla logiker i tillfarten. Observera att även om tiden TO är utmätt innan alla växlingar är klara så avslutas inte logiken förrän den prioriterade gruppen blivit grön.

11	Time-Out	x sek.	sek	**12 sek.**

Används annan räknare än logikens egen räknare (denna räknare anges i Nr. 7) behöver man ej programmera denna parameter, d.v.s. man anger då endast vilken logik som är räknare i parameter Nr. 7.

Copyright © Stockholms Gatu- och Fastighetskontor

Nr. 12 Tillfart.

Här anges busstillfartens signalgruppsnummer så att räknaren vet vilken signalgrupp den tillhör. Räknaren räknar ner när avanmälan sker och den nollställs om TO=0 och tillfarten samtidigt är grön.

12	Tillfart	SGx	SG	**SG1**

Används annan räknare än logikens egen räknare (denna räknare anges i Nr. 7) behöver man ej programmera denna parameter, d.v.s. man anger då endast vilken logik som är räknare i parameter Nr. 7.

Nr. 13 Motsatt räknare.

Här anges räknare för motriktad buss. Används om t.ex. en motriktad och allierade tillfart tas med i prioriteringen och om den motriktade och allierade bussen skall tillåtas förlänga sin egen signalgrupp under den pågående prioriteten.

13	Motsatt räknare	Rx	PR	**R2**

Copyright © Stockholms Gatu- och Fastighetskontor

Prioriteringsbegränsning.

17	Prio.beg. från start	Fx-x sek.	14	
18	Prio.beg. från start	Fx-x sek.	14	
19	Prio.beg. från avslut	Fx-x sek.	15	
20	Prio.beg. från avslut	Fx-x sek.	15	

50	Förhindra egen funktion	x sek.	sek	

<u>**Nr. 17-18** Prio.beg från start.</u>
(Det finns bara plats med 10 logiker i varje cell så därför finns det 2 celler tillgängliga)
Här anges de logiker som skall tillåtas starta inom en angiven tid efter det att logiken har blivit aktiv. Om man inte fyller i någon parameter kommer alla andra logiker att vara förbjudna tills den pågående logiken avslutas.

Prioriteringsbegränsing från start har fått fel utformning.

I dag fungerar begränsningen på så sätt att om man programmerar att logik nr 10 skall begränsas i 30 sekunder kommer logik nr 10 att tillåtas bli aktiv från och med att den aktuella logiken startar och 30 sekunder framöver.

Ny utformning under våren '97. Alla angivna logiker skall prioriteringsbegränsas ett angivet antal sekunder efter det att logiken har blivit aktiv. Alla andra icke angivna logiker kommer att vara prioriteringsbegränsade helt tills logiken är avslutad.

17	Prio.beg. från start	Fx-x sek.	14	**F10-30 sek.**

<u>**Nr. 19-20** Prio.beg från avslut.</u>
(På grund av att det bara är plats för 10 logiker i varje cell så finns det 2 celler tillgängliga)
Här anges de logiker som skall prioriteringsbegränsas en viss angiven tid efter avslutad prioritering. Om man inte fyller i någon parameter kommer alla andra logiker att vara prioriteringsbegränsade under pågående prioritering och alla andra logiker att tillåtas starta efter avslutad prioritering.

Prioriteringsbegränsing från avslut har också fått fel utformning.

I dag fungerar begränsningen på så sätt att om man programmerar att logik nr 10 skall ha begränsning på 30 sekunder kommer **bara** logik nr 10 att tillåtas bli aktiv 30 sekunder efter att den aktuella logiken är avslutad. Alla andra logiker som inte programmeras kommer aldrig någonsin att tillåtas bli aktiva. För att undvika att detta händer måste man programmera alla logiker som inte skall prioriteringsbegränsas med noll sekunder.

19	Prio.beg. från avslut	Fx-x sek.	15	**F10-30 sek., F2,F3,F4,F5,F6,F7,F8,F9-0 sek.**

Copyright © Stockholms Gatu- och Fastighetskontor

Ny utformning under våren '97. Här anges den tid som de angivna logikerna skall prioriteringsbegränsas (förhindras) efter det att den pågående logiken har avslutats. Under pågående prioritet prioriteringsbegränsas (förhindras) alla fientliga logiker d.v.s att alla andra logiker, som ej angivits kommer att tillåtas starta omedelbart efter logikens slut (om lucka och status för dessa är aktiva).

19	Prio.beg. från avslut	Fx-x sek.	15	**F10-30 sek.**

<u>**Nr. 50** Förhindra egen funktion.</u>

Här anges den tid som den egna logiken skall förhindras från att starta pånytt efter avslutad prioritet.

50	Förhindra egen funktion	x sek.	sek	**50 sek.**

Copyright © Stockholms Gatu- och Fastighetskontor

Status.

26	Status; måste grön	SGx	16	
27	Status; kan grön	SGx	17	
28	Status; måste röd	SGx	18	
29	Status; kan röd	SGx	19	
30	Status; annan signal	Anl.x, SGx	In	

Nr. 26 Status; måste grön.

Här anges vilka signalgrupper som måste vara gröna för att logiken skall starta prioriteringen. Observera att om en angiven signalgrupp, som har programmerats att bli grön endast efter anmälan, ej har grönbehov så kommer statusvillkoret inte att uppfyllas.

Kombineras detta villkor med andra typer av villkor (Nr.27-30) så måste samtliga av dessa uppfyllas innan logikens statusvillkor blir aktiva. Om luckan stängs innan statusvillkoren är uppfyllda avslutas logiken. (Observera att grön-blink och grön-gult har status rött och röd-gult har status grönt.)

26	Status; måste grön	SGx	16	**SG1, SG4**

NR. 27 Status; kan grön.

Här anges vilka signalgrupper som kan vara gröna för att logiken skall starta prioriteringen. Det räcker om minst en av de angivna signalgrupperna blir grön för att statusvillkoren skall uppfyllas.

Kombineras detta villkor med andra typer av villkor (Nr.26-30) så måste samtliga av dessa uppfyllas innan logikens statusvillkor blir aktiva. Om luckan stängs innan statusvillkoren är uppfyllda avslutas logiken. (Observera att grön-blink och grön-gult har status rött och röd-gult har status grönt.)

27	Status; kan grön	SGx	17	**SG1, SG4**

Nr. 28 Status; måste röd.

Här anges vilka grupper som måste vara röda för att logiken skall starta prioriteringen.

Kombineras detta villkor med andra typer av villkor (Nr.26-30) så måste samtliga av dessa uppfyllas innan logikens statusvillkor blir aktiva. Om luckan stängs innan statusvillkoren är uppfyllda avslutas logiken. (Observera att grön-blink och grön-gult har status rött och röd-gult har status grönt.)

28	Status; måste röd	SGx	18	**SG2, SG5**

Copyright © Stockholms Gatu- och Fastighetskontor

Nr. 29 Status; kan röd.

Här anges vilka signalgrupper som kan vara röda för att logiken skall starta prioriteringen. Det räcker om minst en av de angivna signalgrupperna är röd för att statusvillkoren skall uppfyllas.

Kombineras detta villkor med andra typer av villkor (Nr.27-30) så måste samtliga av dessa uppfyllas innan logikens statusvillkor blir aktiva. Om luckan stängs innan statusvillkoren är uppfyllda avslutas logiken. (Obsrvera att grön-blink och grön-gult har status rött och röd-gult har status grönt.)

29	Status; kan röd	SGx	19	**SG2, SG5**

NR. 30 Status; annan signal.

Här anges extra villkor för att uppfylla logikens statusvillkor, t.ex. status från en annan signal. Denna parameter kan också användas i speciella fall där funktionens egen statushantering inte räcker till och man vill använda c.blocksprogrammering för status.

Kombineras detta villkor med med funktionens egen statushantering, Nr. 26-29 så måste samtliga av dessa (Nr.27-30) uppfyllas innan logikens statusvillkor blir aktiva. Om luckan stängs innan statusvillkoren är uppfyllda avslutas logiken.

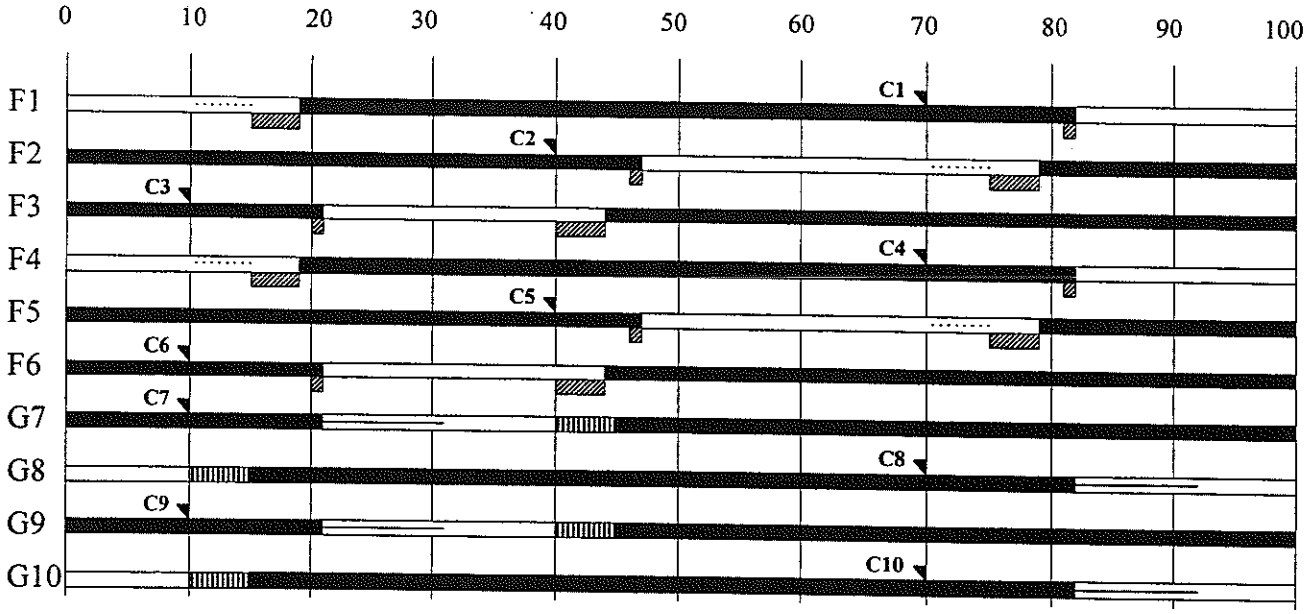
30	Status; annan signal	Anl.x SGx	In	**Anl. 3308 SG4 grön**

Copyright © Stockholms Gatu- och Fastighetskontor

Förlägning och avslut av prioritering.

41	För; SG måste grön	SGx	24	
42				
43	För; SG förlängas	SGx M/Fx sek.	25/20	
44	För; SG till rött garanti	SGx Fx sek.	21	
45				
46	Ny Time-Out SG grön	x sek.	sek	

48	Till rött vid avslut	SGx M/Fx sek.	20/25	
49	Till grönt vid avslut	SGx M sek.	23	



NR. 41 <u>För; SG måste grön.</u>

Här anges signalgrupper som skall förlängas och därför måste vara gröna.

Det är viktigt att signalgrupper som skall förlängas verkligen är gröna innan maxtid eller fråntid ges till dessa. Detta beror på att signalgruppen laddar den ordinarie maxtiden och fråntiden först när gruppen växlat till grönt, vilket medför att om logiken ger en maxtid till signalgruppen medan den fortfarande är röd kommer denna maxtid att bytas ut mot ordinarie maxtid när signalgruppen växlar till grönt.

Logiken väntar på att angivna grupper som har grönbehov och startillstånd, men fortfarande är röda, skall bli gröna.

41	För; SG måste grön	SGx	24	**SG1**

Copyright © Stockholms Gatu- och Fastighetskontor

NR. 43 För; SG förlängas.

Här anges den signalgrupp som skall förlängas och antal sekunder maxtid eller fråntid som den skall tillåtas förlänga.

Det finns två olika sätt att förlänga en signalgrupp. Det ena alternativet är maxtidsförlängning och det andra är fråntidförlängning. I styrapparaten fungerar maxtidsförlängningen på så sätt att gruppen ges en maxtid som räknas ner till noll. På ELC-språk inhiberar man gruppen med en maxtid. Tiden räknas endast ner om det finns någon fientlig start. När maxtiden nollställs (se Nr. 48) t.ex. vid avanmälan växlar signalgruppen därför till rött endast om det finns någon fientlig start. Signalgruppen ligger passivt kvar i grönt tills en fientlig start kommer. När en fientlig start kommer laddas ordinarie fråntid och signalgruppen påbörjar sin växling till rött.

Fråntid eller Past End Greentime är den tid som tas ut efter det att gruppen har fått rödorder. När en signalgrupp laddas med en fråntid ges samtidigt en rödorder och maxtiden nollställs automatisk. Förlängs gruppen med fråntid kommer signalgruppen att växla till rött när fråntiden nollställs oavsett om det finns någon fientlig anmälan eller ej.

Det rekommenderas att maxtidsförlängning används vid alla typer av förlängning, med undantag för de bussar som anmäls när tillfarten mäter ut fråntid, på grund av att busstillfarten blir liggande passivt kvar i grönt vid avanmälan om ingen fientlig grupp har fått startillsånd.

I exemplet laddas SG1 med en maxtid på 60 sekunder. Maxtiden laddas av den första bussen och kommer inte att uppdateras om det kommer en ny buss. Maxtiden bör därför vara så lång så att den räcker från det att luckan för funktionen öppnas och ända till sista-steg. Denna maxtid nollställs (se Nr. 48) när logiken avslutas oberoende av om det är sista-steg, avanmälande detektor eller Time-Out som avslutar logiken.

43	För; SG förlängas	SGx M/Fx sek. 25/20	**SG1 M 60 sek.**

Man kan inte välja båda maxtidsförlängning och fråntidsförlängnng.

NR. 44 För; SG till rött garanti (Se Nr. 34/39) Används ej.

NR. 46 Ny Time-Out SG grön.

Här anges den uppdaterade TO som bussen skall tilldelas när signalgruppen blir grön.

Används lämpligen vid extrafaser om anmälande detektor ligger så nära korsningen att en lång Time-Out är omotiverad. Det är då en uppenbar risk att Time-Out blir noll innan bussens tillfart har blivit grön och det kan då vara nödvändigt att uppdatera TO när busstillfarten blir grön för att säkerställa att bussen hinner genom signalen. Observera att om villkoren för en EF är uppfyllda så växlar den alltid in även om TO=0.

46	Ny Time-Out SG grön.	x sek. sek	**12 sek.**

Copyright © Stockholms Gatu- och Fastighetskontor

NR. 48 Till rött vid avslut.

Här anges de signalgrupper som skall avsluta sin förlängning vid sista-steg, Time-Out=0 eller vid avanmälan. När logiken avslutas efter avanmälan, TO=0 eller sista-steg kan förlängande grupper avslutas med maxtid eller fråntid. Om signalgruppen avslutas med noll sekunder maxtid kommer den, om den har grönbehov, att mäta ut sin ordinarie fråntid. Om ingen fientlig anmälan finns kommer signalgruppen att ligga passivt kvar efter avanmälan. Signalgruppen ställs inte till rött. Om man väljer endast maxtidsavslutning måste sista-steg placeras så att gruppen hinner ta ut sin ordinarie fråntid.

Om tillfarten förlänger fråntid (Se Nr. 43) måste man avsluta förlängningen med en valfri fråntid, som t.ex. kan vara 0 sekunder. Fråntidsavslutning nollställer automatisk maxtiden och signalgruppen börjar omedelbart att "mäta ut" den angivna fråntiden. Om däremot signalgruppen maxtidsförlänger kan man välja om man vill avsluta med en valfri maxtid eller en valfri fråntid. Det är inte möjligt att avsluta en förlängning med t.ex. 5 sekunder maxtid och 5 sekunder fråntid.

Vid fråntidsavslutning nollställs maxtiden automatisk och signalgruppen tvingas till rött direkt med en angiven valfri fråntid oberoende av om det finns en fientlig start eller ej. Ordinarie fråntid tas inte ut. Sista-steg kan därför placeras utan att behöva ta hänsyn till ordinarie fråntid. I det gamla PRIBUSS gav sista-steg en definitiv rödorder till gruppen, så är inte fallet i nya PRIPTS-logiken. Nya PRIBUSS skiljer inte på avanmälan och sista-steg.

48	Till rött vid avslut	SGx M/Fx sek. 20/25	**SG1 M 0 sek.**

Man kan inte välja båda maxtidsförlängning och fråntidsförlängning vid avslut.

NR. 49 Till grönt vid avslut.

Här anges de signalgrupper som skall ges en startorder vid avanmälan, Time-Out=0 eller sista-steg. I en BF behöver vi inte ge start till grupper när logiken avslutas på grund av att start till efterföljande grupper normalt inte är förhindrad. I en EF däremot förhindrar vi normalt ordinarie start till efterföljande läge och då måste vi ge start till dessa grupper efter logikens avslut. Normalt ger man ingen ny tid till signalgrupperna därför att ordinarie maxtid är noll i en samordning.

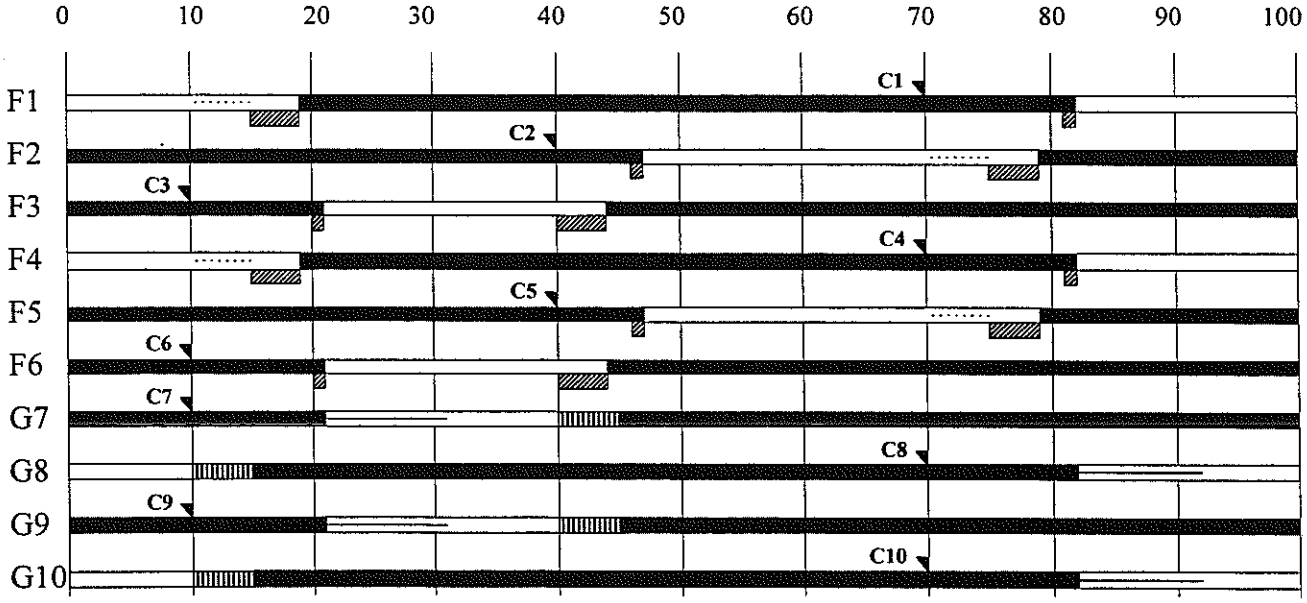
49	Till grönt vid avslut	SGx M sek. 23	**SG2 M 0 sek., SG5 M 0 sek.**

Copyright © Stockholms Gatu- och Fastighetskontor

Avkorta signalgrupper och ge start till signalgrupper.

31	Vx1; SG måste grön	SGx	24	
32				
33	Vx1; SG till rött direkt	SGx M/Fx sek.	25/20	
34	Vx1; SG till rött garanti	SGx Fx sek.	21	
35	Vx1; SG till grönt	SGx Mx sek.	23	

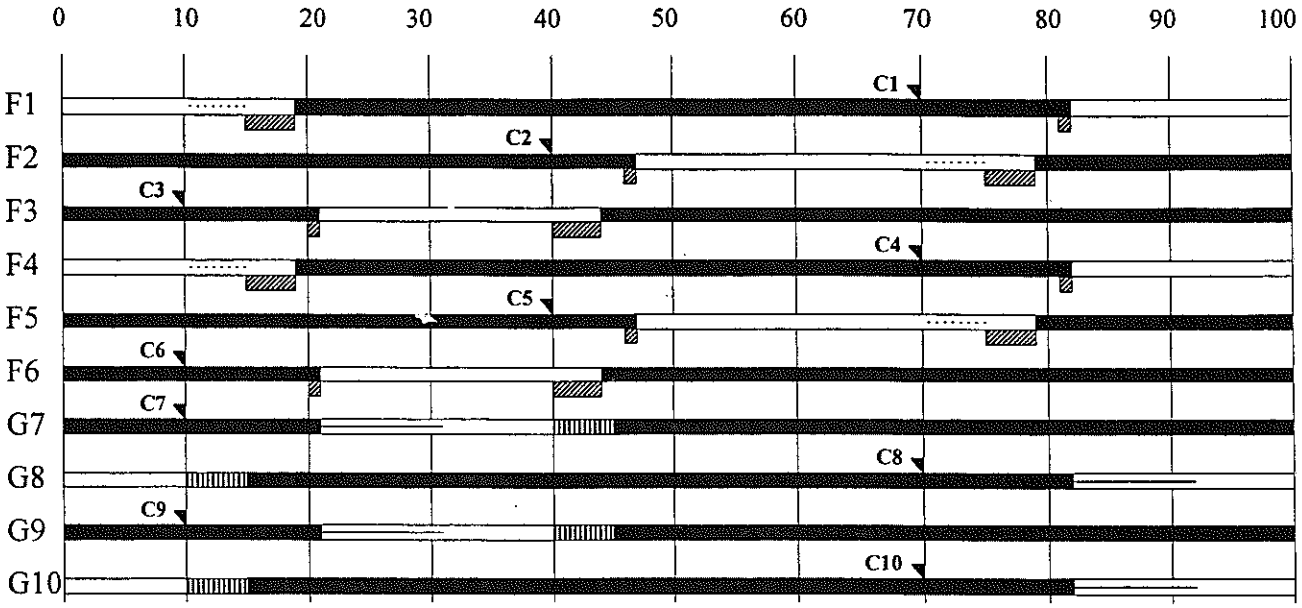
36	Vx2; SG måste grön	SGx	24	
37				
38	Vx2; SG till rött direkt	SGx M/Fx sek.	25/20	
39	Vx2; SG till rött garanti	SGx Fx sek.	21	
40	Vx2; SG till grönt	SGx Mx sek.	23	



Avkortning av signalgrupper används vid ÅTS, AK, EF,DEF, och DAK. Växling1 och Växling2 är identisk lika, men för att få en standard för hur logiken skall användas har det bestämds att Vx1 skall användas, som första växling, vid DEF och DAK och Vx2 skall användas vid ÅTS, EF och AK.

Under ÅTS används avkortning för att stoppa signalgrupper som har fått startorder och för att ge ny start till signalgrupper. Under EF används avkortning normalt för att ge rödorder till grupper som är gröna och starta nya signalgrupper innan ordinarie efterkommande läge startats. Har efterkommande läge vid EF redan startat används avkortning för att stoppa de startade grupperna.

Copyright © Stockholms Gatu- och Fastighetskontor



NR. 31 Vx1; SG måste grön. **NR. 36** Vx2; SG måste grön.

Här anges de grupper som måste vara gröna innan avkortning sker. De grupper som är angivna och saknar grönbehov kommer att förlora sitt startillstånd. Logiken väntar alltså bara på de grupper som har grönbehov och starttillstånd. Växling1 används vid DAK och DEF och Växling2 används vid ÅTS, EF, och AK.

Om en bussanmälan inkommer på t.ex. sekundsteg 60 och luckan samt status stämmer för funktionen AK så skall signalgrupperna F2 och F5 ges en rödorder och signalgrupperna F1, F4, G8 och G10 startas. Innan rödorder och starter ges måste logiken kontrollera att F2 och F5 verkligen är gröna. Om dessa signalgrupper inte är gröna, men har grönbehov, väntar logiken med rödorder och starter tills de definierade signalgrupper blir gröna. Vi kan välja om de skall ges en rödorder direkt med eller utan fråntid, eller ges en rödorder efter utmätt garantitid med eller utan fråntid. Start till efterföljande signal grupper kan ges med eller utan maxtid. I normala fall vid samordnad styrning ger man startorder till efterföljande grupper med maxtid=0.

31	Vx1; SG måste grön	SGx	24	**SG2, SG5**

36	Vx2; SG måste grön	SGx	24	

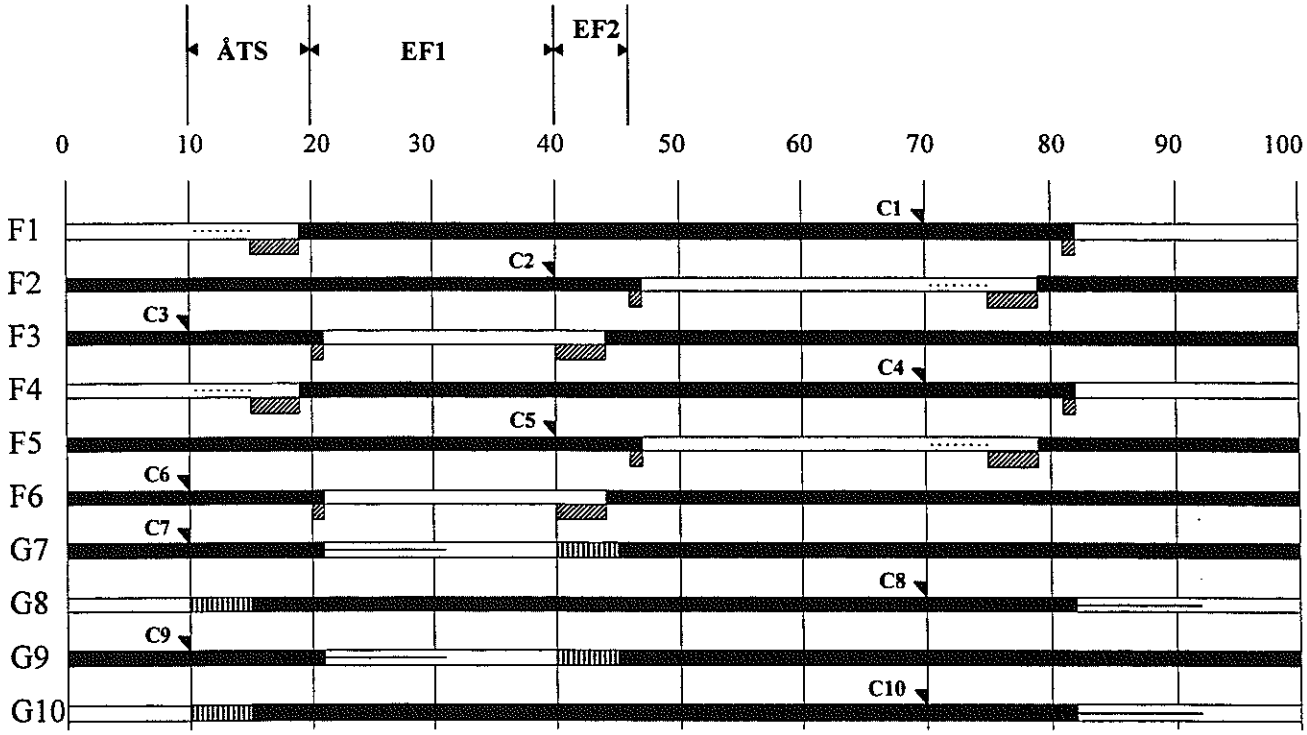
Copyright © Stockholms Gatu- och Fastighetskontor

NR. 33 Vx1; SG till rött direkt. **NR. 38** Vx2; SG till rött direkt.

Här anges de grupper som skall avkortas direkt när mintid är utmätt. Dessa kan ges en valfri maxtid eller fråntid som signalgruppen tillåts ta ut via sina vanliga detektorer efter det att rödordern har kommit. Observera att logiken omedelbart ställer angivna grupper till rött, jämförbart med en s.k. stoppuls. Maxtiden nollas automatisk och eventuellt angivna fråntiden tas ut efter rödordern. Normalt sätter man fråntiden vid denna funktion till 0 sekunder. Växling1 används vid DAK och DEF och Växling2 används vid ÅTS, EF, och AK.

Vid ÅTS -funktionen skall de signalgrupper som fått startorder förhindras att starta, därför ges en direkt rödorder (fråntid=0) till signalgrupperna F3, F6, G7 och G10 utan att behöva vänta på att några grupper skall bli gröna. Denna rödorder förhindrar då start till dessa signalgrupperna och därmed även möjligheten för någon av dessa att "greppa" starten före den prioriterade ÅTS-gruppen. Signalgrupperna F1 och F4 ges en startorder med valfri maxtid. Förlängningen definieras i förlängningsmodulen (Nr. 41-44) som är beskriven tidigare.

Det händer även ofta att vi vill använda funktionen EF så länge som möjligt. T.ex. är det inte ovanligt att vi väljer att extrafasen skall tillåtas efter det att start till signalgrupperna F2 och F5 har kommit (se figur nedan). Det enda kravet är då att F2 eller F5 inte har hunnit bli gröngula.



I den tidigare PRIBUSS funktionen valde vi att definiera status som B1 och B2. I nya PRIPTS-logiken är vi tvungna att använda två extra faser. En extrafas som tillåts när F3, F6 etc är gröna och en extrafas med ett annat statusvillkor (EF2 i fig. ovan) när F2 och F5 redan fått sin start. Den senare extrafasen EF2 blir då jämförbar med en ÅTS. (forts. nästa sida)

Copyright © Stockholms Gatu- och Fastighetskontor

Vi ger därför direkt rödorder till F2 och F5 utan att kontrollera om dessa grupper är gröna. Rödordern tar då bort startordern till dessa grupper samtidigt som de prioriterade grupperna i extrafasen får startorder och möjligheten att någon av F2 eller F5 skall kunna "greppa" starten före den prioriterade EF-gruppen förhindras. Startorder till de prioriterade grupperna ges i parameter Nr. 40.

33	Vx1; SG till rött direkt	SGx M/Fx sek. 25/20	

38	Vx2; SG till rött direkt	SGx M/Fx sek. 25/20	**SG2 F 0 sek., SG5 F 0 sek.**

Man kan inte välja båda maxtidsförlängning och fråntidsförlängning.

<u>**NR. 34** Vx1; SG till rött garanti. **NR. 39** Vx2; SG till rött garanti.</u>

Alla grupper som anges här är beroende av varandra och kommer att växla till rött (d.v.s. få rödorder) samtidigt. Den grupp som har längst mintid eller garantitid är dimensionerande. Övriga grupper "hänger" då kvar i grönt (om de inte är programmerad "till rött efter upphört grönbehov") tills denna tid är utmätt. Observera att garantitiden alltid är trafikstyrd och gäller för alla tidplaner. Grupperna kan ges en valfri fråntid som dessa tillåts ta ut via sina vanliga detektorer efter det att rödorder har kommit. Jmfr. skillnaden med grupperna som anges i Nr.33 (38) som omedelbart ställer angivna grupper till rött. Växling1 används vid DAK och DEF och Växling2 används vid ÅTS, EF, och AK.

Denna extra finess med den nya PRIPTS-logiken gör att signalgrupper som har kort mintid eller garantitid kan ligga passiva kvar i grönt fast de har fått en rödorder med noll sekunder fråntid. Signalgrupperna F3, F6, G7 och G9 skall ställas till rött och F1 och F4 skall ges en extrafas. Vi vill avkorta F3,F6, G7 och G9 så snabbt som möjligt och ge start till F1 och F4, men vi vill också att F3 och F6 skall ligga passiva kvar tills mintiden i G7 och G9 är utmätt. Parameter Nr.39 (även 34) "SG till rött garanti" fungerar på så sätt att alla angivna signalgrupper ligger kvar i grönt tills <u>alla</u> har mätt ut sin mintid och garantitid. Det vill säga att om F3, F6 G7 och G9 är angivna i denna parameter kommer F3 och F6 att ligga kvar passivt i grönt tills mintiden är utmätt i G7 och G9. Om F3 och F6 har ordinarie fråntid kommer grupperna inte att mäta ut fråntid om vi anger noll sekunder i denna parameter.

34	Vx1; SG till rött garanti	SGx Fx sek. 21	

39	Vx2; SG till rött garanti	SGx Fx sek. 21	**SG3, SG6, SG7, SG9 F 0 sek.**

Copyright © Stockholms Gatu- och Fastighetskontor

NR. 35 Vx1; SG till grönt . **NR. 40** Vx2; SG till grönt .

Här anges de grupper som skall starta i nästa läge. Växling1 används vid DAK och DEF och Växling2 används vid ÅTS, EF, och AK.

Parameter Nr.35 används för att ge startorder till de signalgrupper som skall få tidigare start i en dubbel avkortning (DAK) eller en dubbel avkortning med extra fas (DEF). Parameter Nr. 40 används för att ge start till de prioriterade signalgrupperna i funktionerna ÅTS, EF, AK, DAK och DEF. Normalt ger man ingen maxtid till signalgrupperna beroende på att grupperna vid samordnad styrning ligger passivt kvar i grönt tills fientlig start kommer. (Eventuell maxtid "tas ut" efter det att fientlig start kommer.)

35	Vx1; SG till grönt	SGx Mx sek.	23	

40	Vx2; SG till grönt	SGx Mx sek.	23	**SG1 M 0 sek., SG4 M 0 sek.**

Copyright © Stockholms Gatu- och Fastighetskontor

Luckor och sista-steg.

16	Lucka	Px;Cx+x sek.-Cx+x sek. 2-3	
	Lucka NyBuss	Px;Cx+x sek.-Cx+x sek. 4-5	
	Lucka Motriktad	Px;Cx+x sek.-Cx+x sek. 6-7	

47	Sista-Steg	Px;Sx+x sek. (28,PRP16)	

<u>**NR. 16** Lucka, Lucka NyBuss, Lucka Motriktad **NR. 47** Sista-steg</u>

I tidigare versioner av bussprioritering fanns bara en tidlucka som tillsammans med status-villkoret bestämde om logiken skulle tillåtas starta och om ytterligare en ny buss skulle tillåtas.

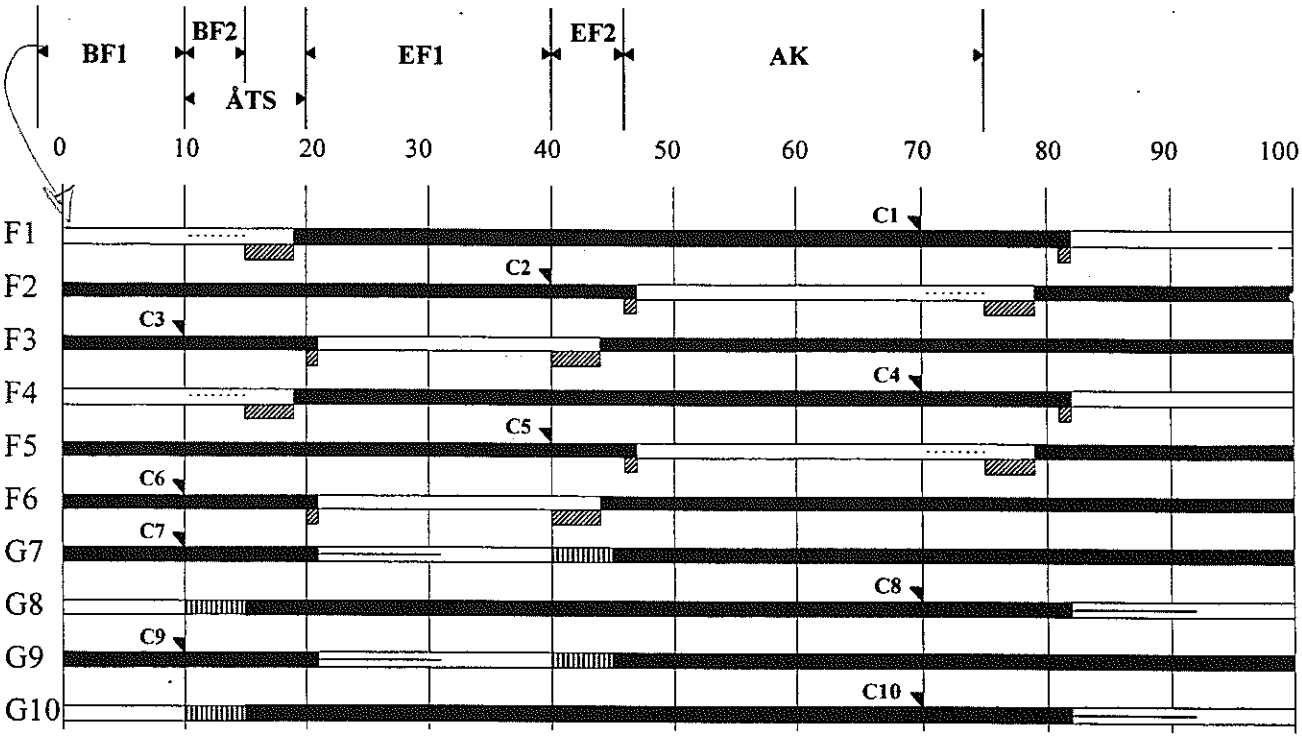
I nya PRIPTS-logiken har vi tillgång till tre luckor som beskriver var i omloppet logiken skall tillåtas starta, hur länge en ny buss skall tillåtas förlänga samma funktion och om en ny allierad buss i motriktad körfält skall tillåtas förlänga sin gröntid; **Lucka**, **NyBuss** och **NyBussMot**.

Lucka definerar var en viss logik tillåts starta. Huvudprincipen i nya PRIPTS-logiken är att hela omloppet skall täckas av luckor. Om det finns ett glapp mellan två luckor och en buss anmäls i detta glapp så kommer ingen logik att starta. Om luckan för en viss logik är öppen och status inte stämmer när bussen anmäls så ligger logiken hela tiden och väntar tills status stämmer så länge luckan är öppen. Om luckan stängs innan status stämmer kommer logiken att avslutas. Ingen annan logik kommer att ta över och bussen prioriteras ej. Om två luckor överlappar varandra kommer den prioritet som först får rätt status att verkställas.

Det finns fall där vi inte vill använda någon PRIBUSS- funktion därför att den inte behövs t.ex. om bussen kommer i början av gröntiden och TO är kortare än tillgänglig gröntid. I gamla PRIBUSS-funktionen fanns en möjlighet att bevaka gröntiden så att inte fientliga funktioner kunde ta över. I den nya PRIPTS-logiken kan vi använda vanlig BF utan att sätta grupper till rött eller grönt efter avslutat prioritering. Mer om detta lite senare.

Copyright © Stockholms Gatu- och Fastighetskontor

Förslag till val av Lucka:



I vårt exempel vill jag föreslå att man väljer två bussförlängningar; BF1 om bussen anmäls innan fientlig start och BF2 om tillfarten mäter ut fråntid. Kommer bussen i början på gröntiden hinner den ut utan prioritet. Det är heller inte nödvändigt att bevaka mot fientliga prioriteter. Luckan behöver inte öppnas för i slutet av gröntiden. Det finns inte någon inbyggd automatik i den nya PRIPTS-logiken som känner av om tillfarten mäter fråntid eller bara ligger passiv. Projektören måste bestämma vilken typ av förlängning som skall användas. En logik kan inte båda förlänga maxtid och fråntid. BF1 skall förlänga maxtid och BF2 skall förlänga fråntid. Vid avanmälan vill jag föreslå att man nollställer tillfartens maxtid i BF1 vilket betyder att sista-steg måste placeras 5 sekunder tidigare på grund av att ordinarie fråntiden tas ut efter sista-steg. I BF2 kan fråntiden nollställas och tillfarten växlar från direkt.

Luckan för ÅTS öppnas direkt efter att fientlig start har kommit. Status avgör om ÅTS eller BF2 skall användas. Fientliga signalgrupper får inte ha blivit rödgula eller gröna. Det första logiken gör är att stoppa fientliga grupper från att växla. Detta görs med en "rött direkt" order till signalgrupperna.

EF1 avkortar ordinarie läget efter bussläget och "kilar in " bussläget mellan läge 2 och läge 3. Vi väljer två extrafaser därför att det finns möjlighet att "kila in" bussläget efter att läge 2 har växlat bort och innan läge 3 har växlat till grönt (rödgult). Status är helt olika i dom två fallen. Om man inte vill använda ÅTS kan man välja att öppna luckan för EF tidigare. Status är att någon av F3, F6, G7 eller G9 är grön, kombinerat med parameter Nr. 36 "SG måste grön" för F3, F6, G7 och G9. Logiken väntar då på att de andra grupper som har grönbehov och startillstånd skall bli gröna innan avkortning sker.

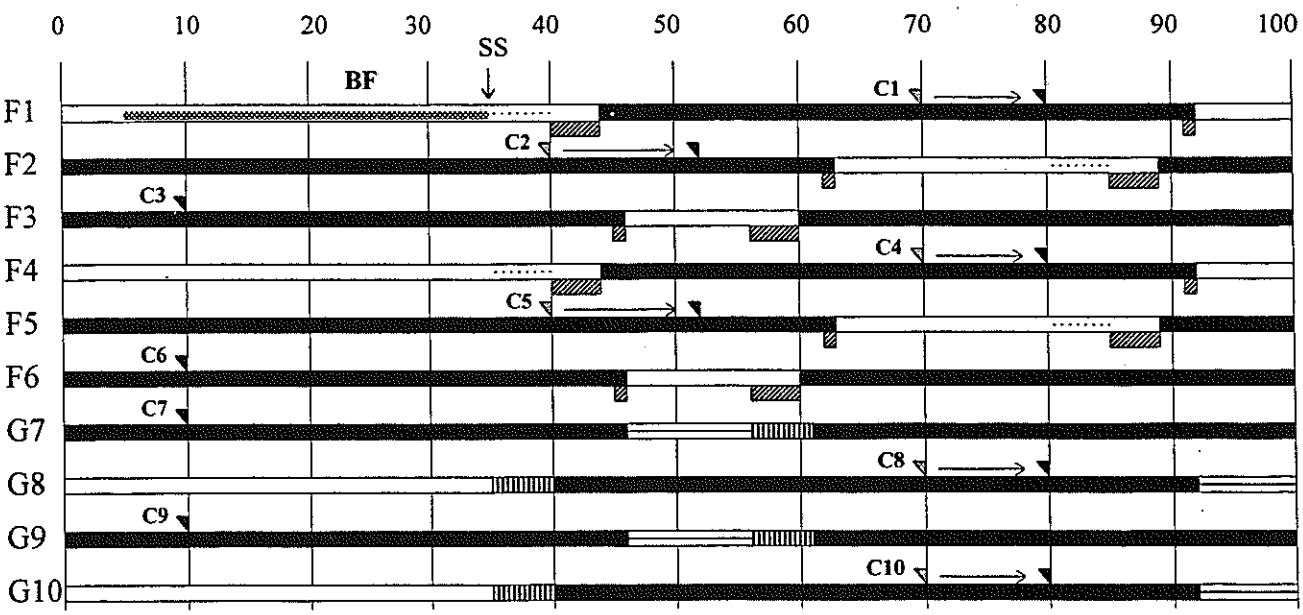
Copyright © Stockholms Gatu- och Fastighetskontor

AK verkställer en tidigare start till bussläget . Om inte EF2 önskas kan luckan för AK öppnas när läge 2 växlar bort. Status är F2 eller F5 är grön beroende på om det är F1 eller F4 som skall prioriteras och logiken väntar tills dess att status stämmer innan avkortning sker.

Det finns också funktionerna dubbel avkortning med extrafas, DEF och dubbel avkortning, DAK som beskrivs i exemplen.

Sista-steg .

Sista-steg bestäms av filosofien att ingen signalgrupp får "hoppas över". Logiken måste avslutas så tidigt att alla efterföljande grupper hinner bli gröna inom i princip en omloppscykel efter prioriteringens start. Efterföljande starter måste alltså flyttas så att alla grupper hinner bli gröna. Om logiken aktiveras t.ex. precis före starterna C3, C6, C7 och C9 kan inte dessa starter flyttas i nästa omlopp. Om logiken skall flytta dessa starter kommer dessa att flyttas direkt efter att logiken har aktiverats och inte i omloppet efter. Det betyder att dessa starter ges på ordinarie sekundsteg i omloppet efter oavsett om läget före har hunnit blivit grönt eller ej. Om logiken startar efter C3, C6 etc. kan dessa starter flyttas i omloppet efter. Då måste alla grupper hunnit blivit gröna innan C2 och C5 kommer, Comprende!! I exemplet i figuren nedan måste alltså alla signalgrupper hinna bli gröna mellan sista-steg och C3, C6, C7 samt C9.



Det är viktigt att observera att om maxtidsförlängning och om man avslutar prioriteringen med nollställning av maxtiden, så kommer ordinarie fråntid att tas ut efter Sista-steg.

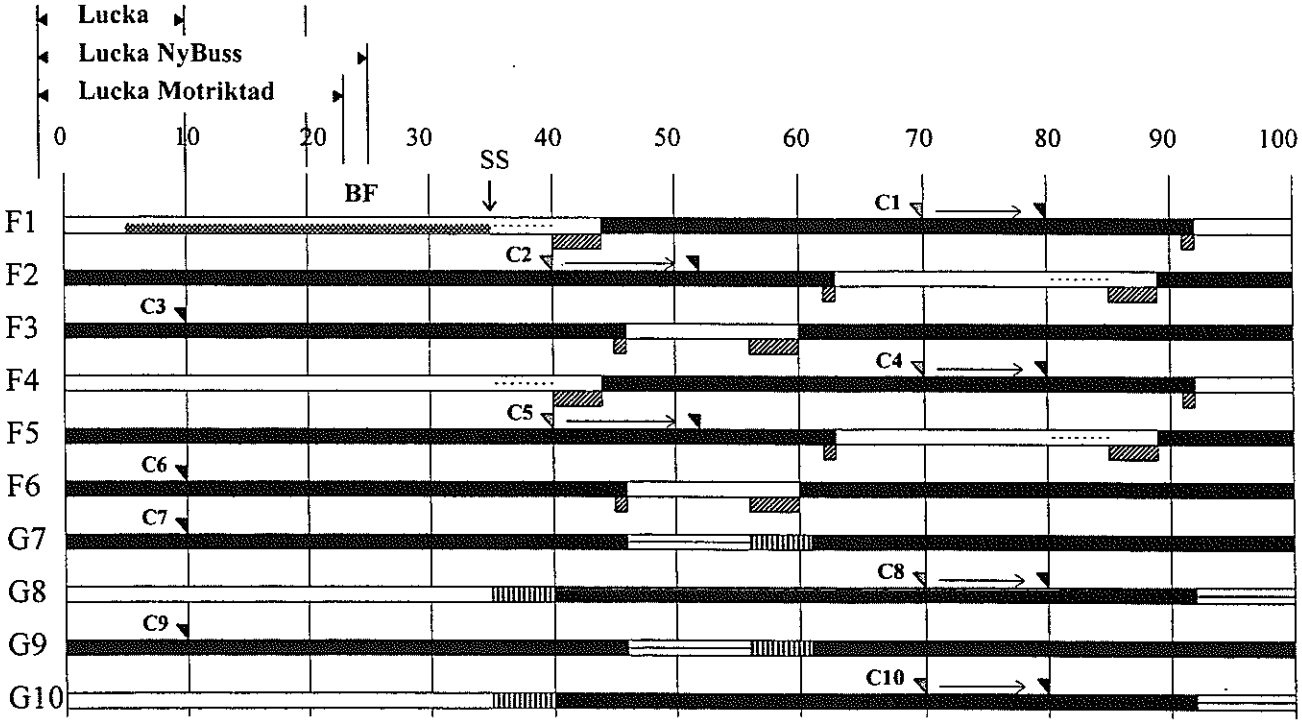
Copyright © Stockholms Gatu- och Fastighetskontor

Lucka NyBuss.

När man har bestämt sista-steg är det ingen konst att definiera hur länge nya bussar skall få ta upp förlängning. Lucka NyBuss öppnas samtidigt som Lucka och stängs före sista-steg. Sista-steg minus körtiden är en lämplig formel.

Lucka NyBussMotriktad.

Det samma gäller för hur länge en allierad buss skall tillåtas förlänga; sista-steg minus körtid i allierad bussttillfart.



16	Lucka	Px;Cx+x sek.-Cx+x sek. 2-3	**P1; C1+18-C3**
	Lucka NyBuss	Px;Cx+x sek.-Cx+x sek. 4-5	**P1; C1+18-C2+2**
	Lucka Motriktad	Px;Cx+x sek.-Cx+x sek. 6-7	**P1; C1+18-C2+4**

47	Sista-Steg	Px;Sx+x sek. (28,PRP16)	**P1; C2+14**

Sista-steg: C1+14; 54

Kommihåg: Inga luckor får öppnas eller stängas i sekundsteg 0. Gäller även Sista-steg.

Copyright © Stockholms Gatu- och Fastighetskontor

Flyttta starter och förhindra starter.

21	Lucka Flytt1	Px;Cx+x sek.-Cx+x sek. 8-9	
	Lucka Flytt2	Px;Cx+x sek.-Cx+x sek. 10-11	
22	Flytt1, Starter	Px;Cx+x sek. 12	
23	Flytt1, Starter	Px;Cx+x sek. 12	
24	Flytt2, Starter	Px;Cx+x sek. 13	
25	Flytt2, Starter	Px;Cx+x sek. 13	

Nr. 21 Lucka Flytt1, Lucka Flytt2 **Nr. 22-23** Flytt1, Starter **Nr. 24-25** Flytt2, Starter

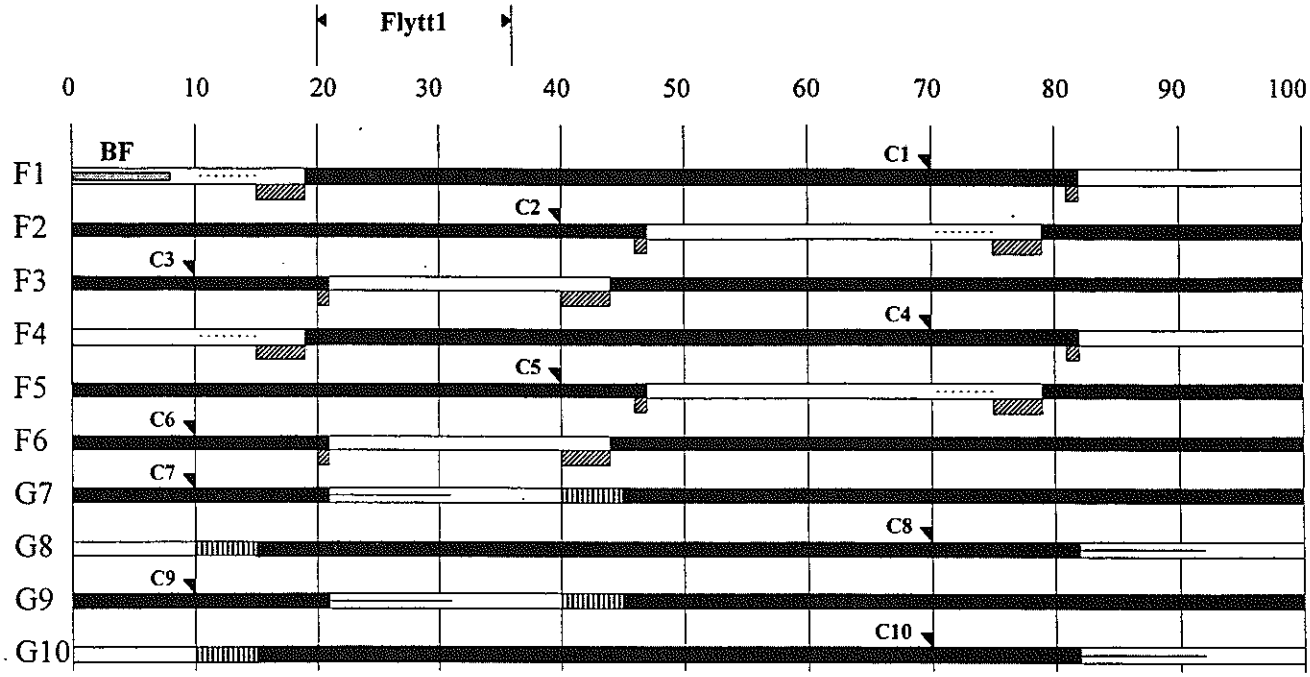
Flyttning av starter har fått en ny utformning jämfört med gamla PRIBUSS. Det finns ingen särskild tumregel för hur efterföljande starter skall flyttas. Huvudkravet är att alla signalgrupper efter prioriteringens avslut hinner bli gröna inom i princip en omloppscykel efter det att luckan öppnades.

Observera att alla starter oavsett om det är ordinarie starter, flyttade starter från PRIPTS-logiken eller starter som ges direkt från PRIPTS-logiken (i t.ex. en EF) har samma prioritet. Det finns ingen inbyggd logik i styrapparaten som ger starter från PRIPTS-logiken högre prioritet än ordinarie starter. Det finns heller ingen inbyggd logik som kontrollerar att flyttade starter verkställs. Om logiken flyttar starter till efterföljande signalgrupper och det kommer ordinarie konflikterande starter innan signalgrupperna hinner bli gröna så är det fientlighets-matrisen som avgör vilka signalgrupper som skall bli gröna. Det är viktigt att projektören kontrollerar att inga konflikterande startorder ges.

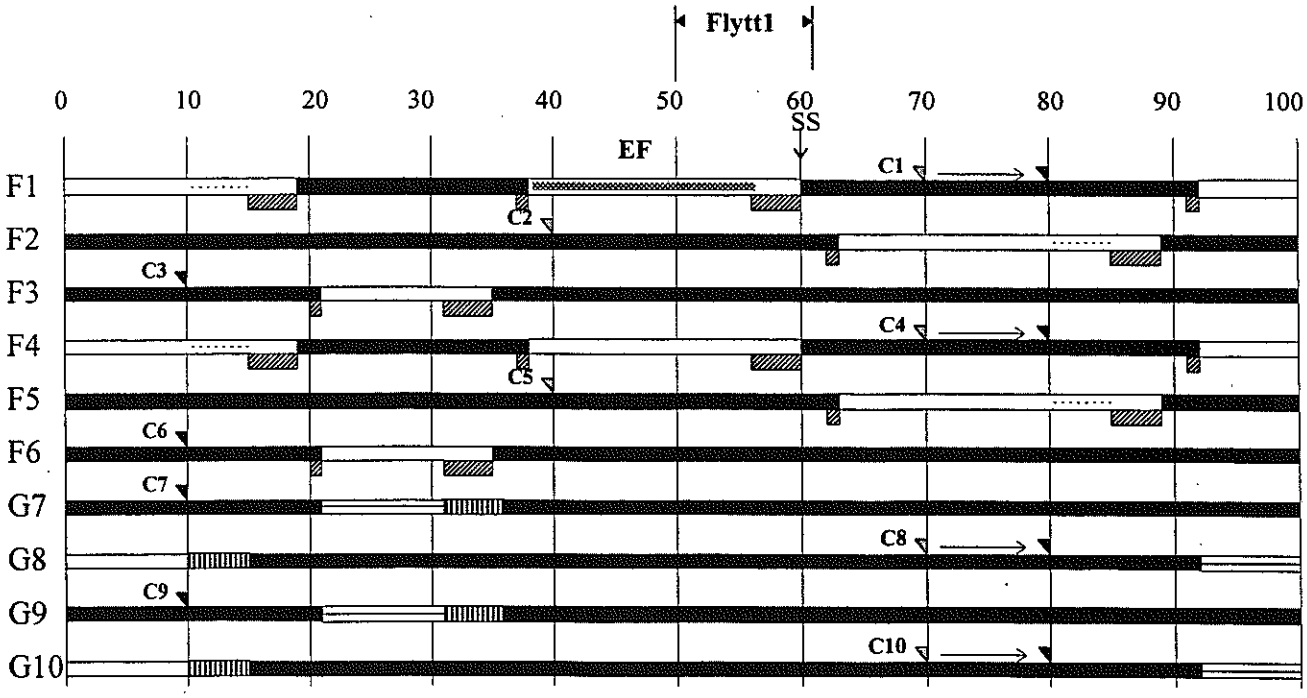
I nya PRIPTS-logiken finns det två luckor som hjälper oss att flytta starter. Om logiken avslutas när Flytt1 är aktiv kan vissa angivna starter flyttas ett antal sekunder och vidare om logiken avslutas när Flytt2 är aktiv har man möjlighet att ändra vilka starter som skall flyttas samt längden på flyttningarna. Med avslut menas när logiken ger rödorder till prioriterade grupper och ger startorder till efterföljande grupper (se Nr.48, 49). Om man bara användar en flyttning så bör Flytt1 utnyttjas. Vid mindre prioriteter behöver man inte använda någon flyttfunktion alls.

Om en start flyttas 10 sekunder från ordinarie start på sekundsteg 50 till sekundsteg 60 adderas siffran 50 med 10 i minnet vilket betyder att man inte behöver förhindra ordinarie startorder. När startordern är exekverad byts siffran 60 ut mot ordinarie sekundsteg. Alla efterföljande starter kan alltså bara flyttas en gång. Det är inte möjligt att tidigarelägga starter med flyttfunktionen. Hur man förhindrar starter beskrivs i stycket nedan med samma namn.

Copyright © Stockholms Gatu- och Fastighetskontor

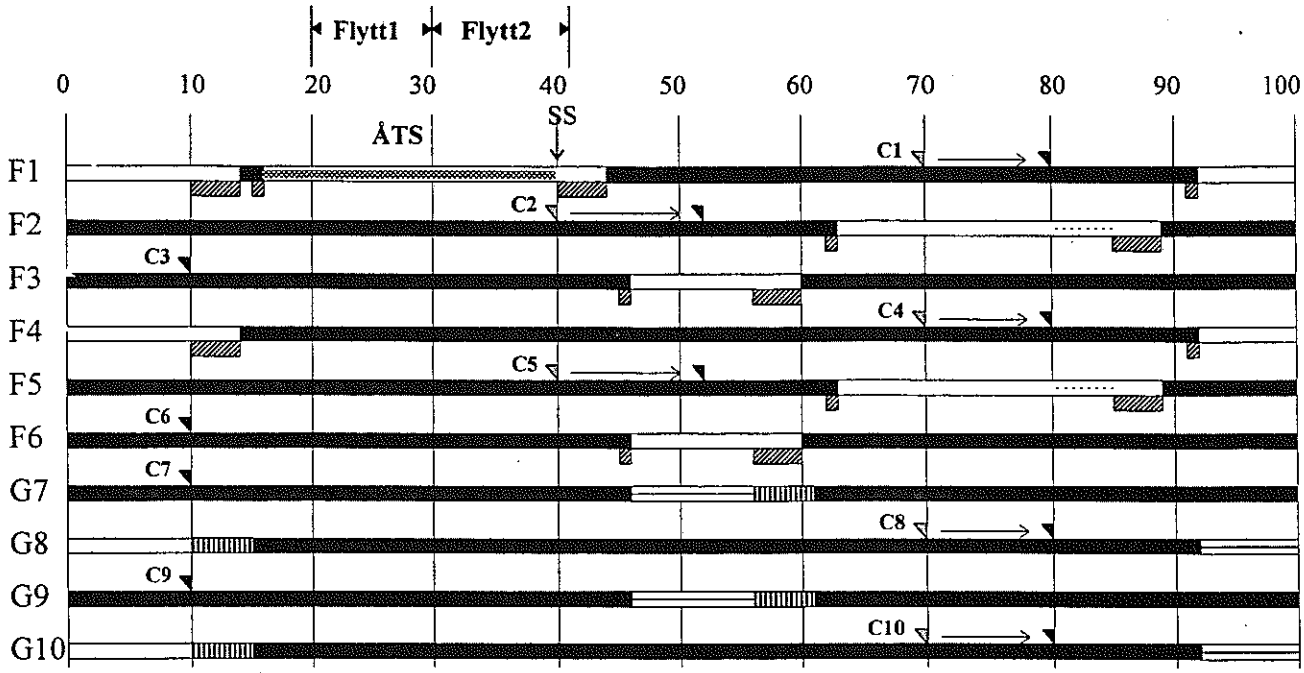


I exemplet visas en BF med maxtidsförlängning som avslutas innan luckan Flytt1 har öppnat. Inga starter flyttas.



Exemplet visar en EF med maxtidsförlängning som avslutas när luckan Flytt1 är aktiv. C1, C4, C8 och C10 flyttas 10 sekunder.

Copyright © Stockholms Gatu- och Fastighetskontor



Exemplet visar en ÅTS med maxtidsförlängning som avslutas när luckan Flytt2 är aktiv. C2 och C5 flyttas 12 sekunder och C1, C4, C8 och C10 flyttas 10 sekunder. Observera att luckan för Flytt2 stängs en sekund efter sista-steg för att säkerställa att luckan verkligen är aktiv när sista-steg kommer.

21	Lucka Flytt1	Px;Cx+x sek.-Cx+x sek. 8-9	**P1; C3+10-C3+20**
	Lucka Flytt2	Px;Cx+x sek.-Cx+x sek. 10-11	**P1; C3+20-C2+1**
22	Flytt1, Starter	Px;Cx+x sek. 12	**P1; C2+7,C5+7,C1+5,C4+5,C8+5,C10+5**
23	Flytt1, Starter	Px;Cx+x sek. 12	
24	Flytt2, Starter	Px;Cx+x sek. 13	**P1; C2+12,C5+12,C1+10,C4+10,C8+10,C10+10**
25	Flytt2, Starter	Px;Cx+x sek. 13	

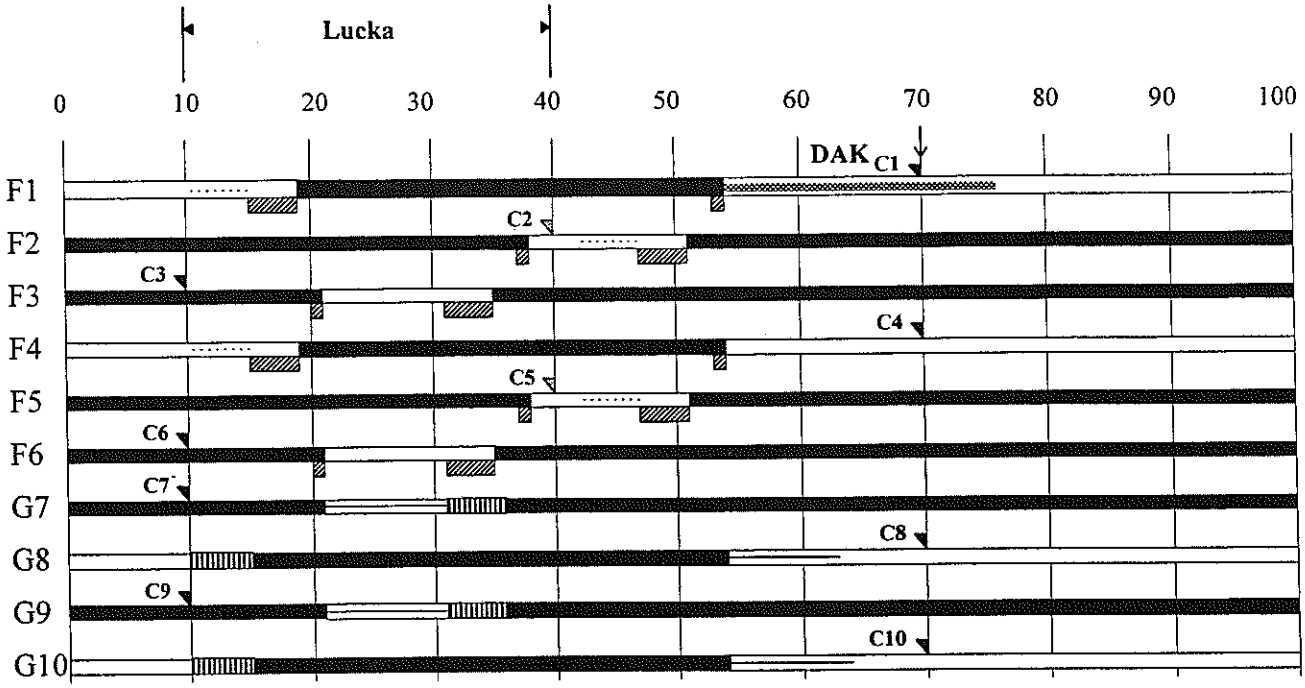
Flytt1; C3+10-C3+25; 20-35
Flytt2; C3+25-C2+15; 35-55

Om logiken avslutas mellan sekundsteg 20 och 35 skall C2 och C5 flyttas 7 sekunder och C1, C4, C8 och C10 flyttas 5 sekunder. Om logiken avslutas mellan sekundsteg 35 och 55 skall C2 och C5 flyttas 12 sekunder och C1, C4, C8 och C10 flyttas 10 sekunder.

Copyright © Stockholms Gatu- och Fastighetskontor

Förhindra Starter.

Om man flyttar starter behövs inte den ordinarie starten förhindras. Det finns ändå tillfällen då ordinarie starter måste förhindras för att logiken skall fungera på ett tillfredsställande sätt. Som tidigare förklarat är det ingen skillnad mellan ordinarie starter och starter som ges från PRIPTS-logiken. Det gäller båda flyttade starter och grönorder till signalgrupper som skall startas i t.ex. i en extrafas.



Exemplet visar en dubbel avkortning där starten till F2 och F5 inte behövs. (I just detta fall behöver man inte förhindra C2 och C5.)

21	Lucka Flytt1	Px;Cx+x sek.-Cx+x sek. 8-9	
	Lucka Flytt2	Px;Cx+x sek.-Cx+x sek. 10-11	
22	Flytt1, Starter	Px;Cx+x sek. 12	**P1; C2=Stp, C5+255**
23	Flytt1, Starter	Px;Cx+x sek. 12	
24	Flytt2, Starter	Px;Cx+x sek. 13	
25	Flytt2, Starter	Px;Cx+x sek. 13	

Observera att det inte behövs någon Flytt-lucka. Man kan använda båda "255" och "Stp" för att beskriva att C2 och C5 skall förhindras.

Copyright © Stockholms Gatu- och Fastighetskontor

Exempel. Beskrivning av generella parametrar och räknare.

Vi använder exemplet på sidan 6 som grund för beskrivning. Tillfart F1 och F4 är prioriterade tillfarter. Time-Out i F1 är 12 sekunder och TO i tillfart F4 är 17 sekunder.

Generella parameter, räknare och status är beskrivet och kräver ingen ytterligare förklaring än att vi reserverar logikerna 1-9 för tillfart F1 och därmed blir logik 10 den första lediga logik för tillfart F4. Vi lägger alltså motsatt räknare på logik 10 (Parameter Nr. 13).

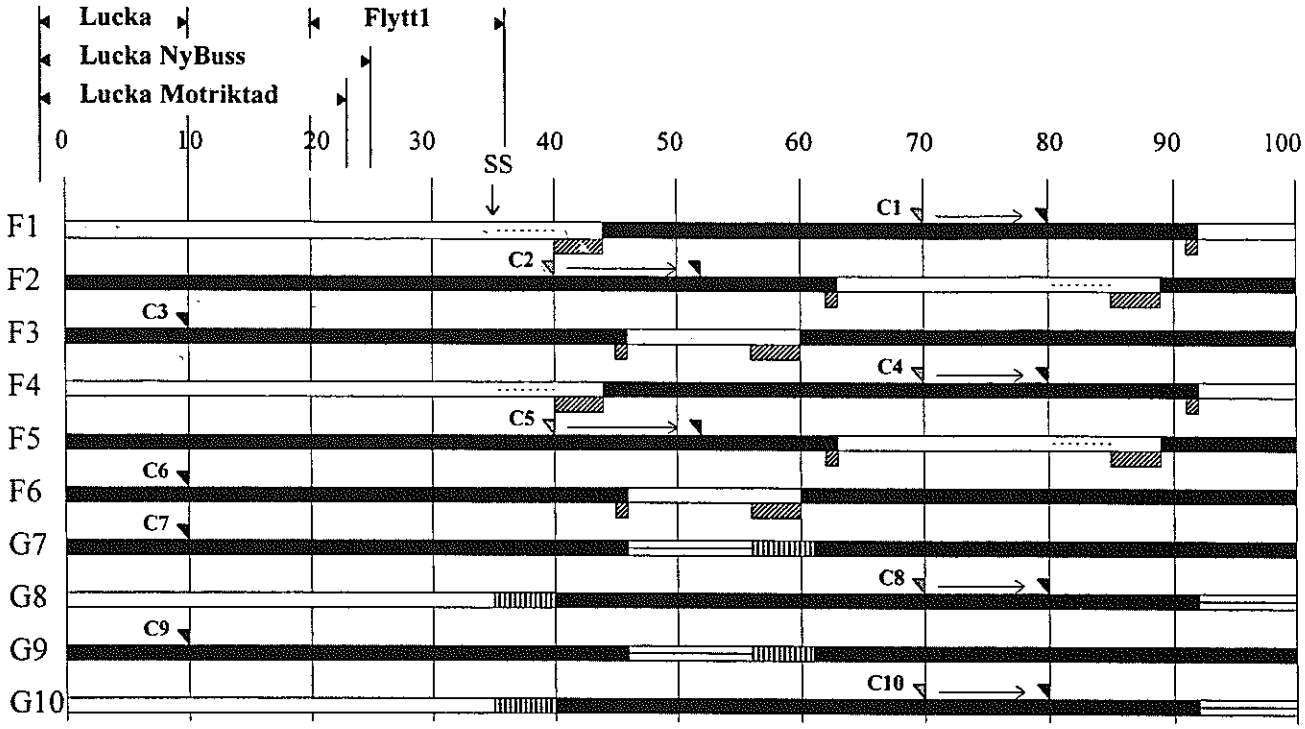
1	Typ av funktion	BF,ÅTS.....	1,2..	**BF**
2				
3	Aktiv i tidplan	Px	1/27	**P2**
4				
5				
6	Förlängande detektor	SGx Mi Fi sek.	DP	**SG1 M 0.1 F 0.1 sek.**
7	Räknare	Rx	PR	**R1**
8	Anmälande detektor	Det x	In	**BD 1-1**
9	Avanmälande detektor	Det x	In	**BD 1-2**
10				
11	Time-Out	x sek.	sek	**12 sek.**
12	Tillfart	SGx	SG	**SG1**
13	Motsatt räknare	Rx	PR	**R10**

För funktion BF2 (fråntidsförlängning), ÅTS etc använder vi samma TO, anmälande och avanmälande detektor som logik nr.1 (BF1, maxtidsförlängning) och därför behövs inte parameter 8-12 fyllas i;

1	Typ av funktion	BF,ÅTS.....	1,2..	**ÅTS**
2				
3	Aktiv i tidplan	Px	1/27	**P2**
4				
5				
6	Förlängande detektor	SGx Mi Fi sek.	DP	**SG1 M 0.1 F 0.1 sek.**
7	Räknare	Rx	PR	**R1**
8	Anmälande detektor	Det x	In	
9	Avanmälande detektor	Det x	In	
10				
11	Time-Out	x sek.	sek	
12	Tillfart	SGx	SG	
13	Motsatt räknare	Rx	PR	**R10**

Copyright © Stockholms Gatu- och Fastighetskontor

Exempel BF1 (Maxtidsförlängning)

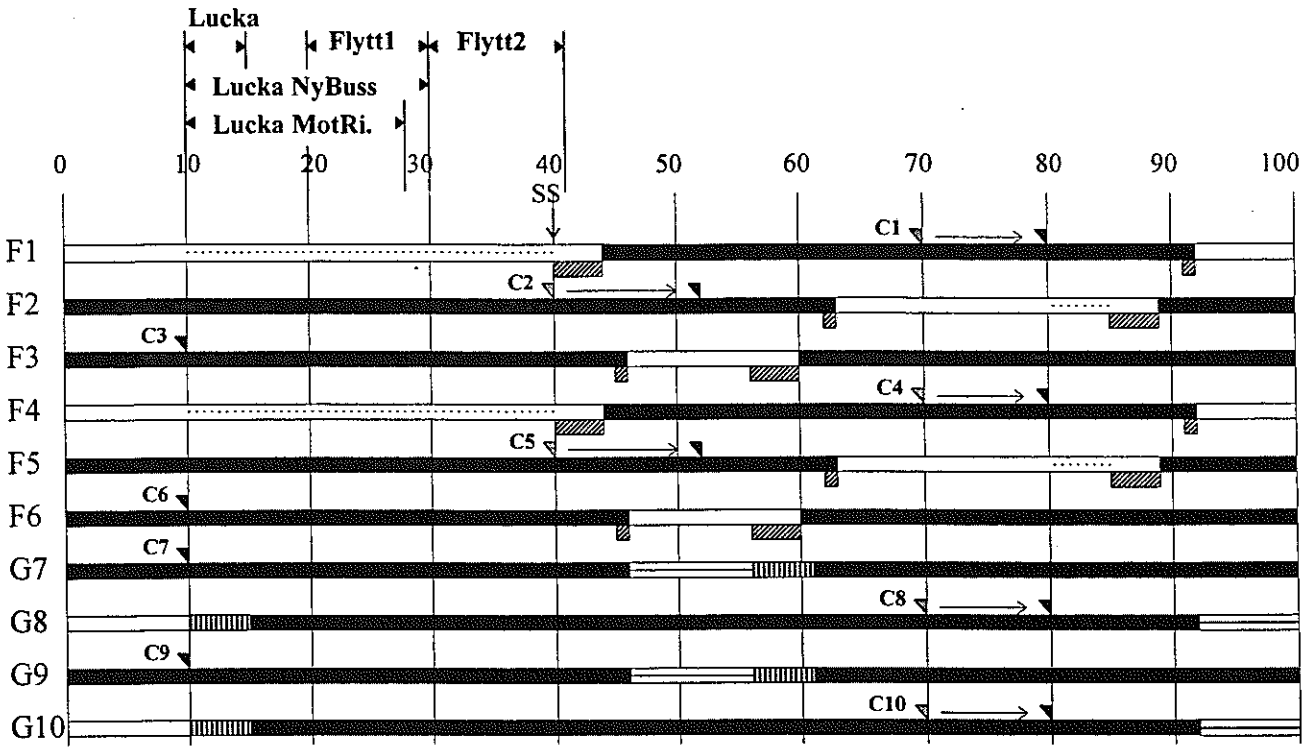


1	Typ av funktion	BF,ÅTS.....	1,2..	**BF1**
6	Förlängande detektor	SGx Mi Fi sek.	DP	**SG1 M 0.1 F 0.1 sek.**
16	Lucka	Px;Cx+x sek.-Cx+x sek.	2-3	**P1; C1+28-C3**
	Lucka NyBuss	Px;Cx+x sek.-Cx+x sek.	4-5	**P1; C1+28-C3+15**
	Lucka Motriktad	Px;Cx+x sek.-Cx+x sek.	6-7	**P1; C1+28-C3+13**
21	Lucka Flytt1	Px;Cx+x sek.-Cx+x sek.	8-9	**P1; C3+10-C3+26**
22	Flytt1, Starter	Px;Cx+x sek.	12	**P1; C2,C5+12 sek C1,C4,C8,C10+10 sek**
26	Status; måste grön	SGx	16	**SG1**
41	För; SG måste grön	SGx	24	**SG1**
43	För; SG förlängas	SGx M/Fx sek.	25/20	**SG1 M 50 sek**
47	Sista-Steg	Px;Sx+x sek.	(28,PRP16)	**P1; C3+25**
48	Till rött vid avslut	SGx M/Fx sek.	20/25	**SG1 M 0 sek**

Observera att maxstidsförlängning används och maxtiden nollställs vid logikens avslut vilket medför att signalgrupperna växlar till rött med ordinarie fråntid vid avslut. Om man använder fråntidsavslutning nollas maxtiden automatisk och signalgrupperna växlar normalt direkt till rött vid avslut oberoende av om det finns fientlig start.

Copyright © Stockholms Gatu- och Fastighetskontor

Exempel BF2 (Fråntidsförlängning)

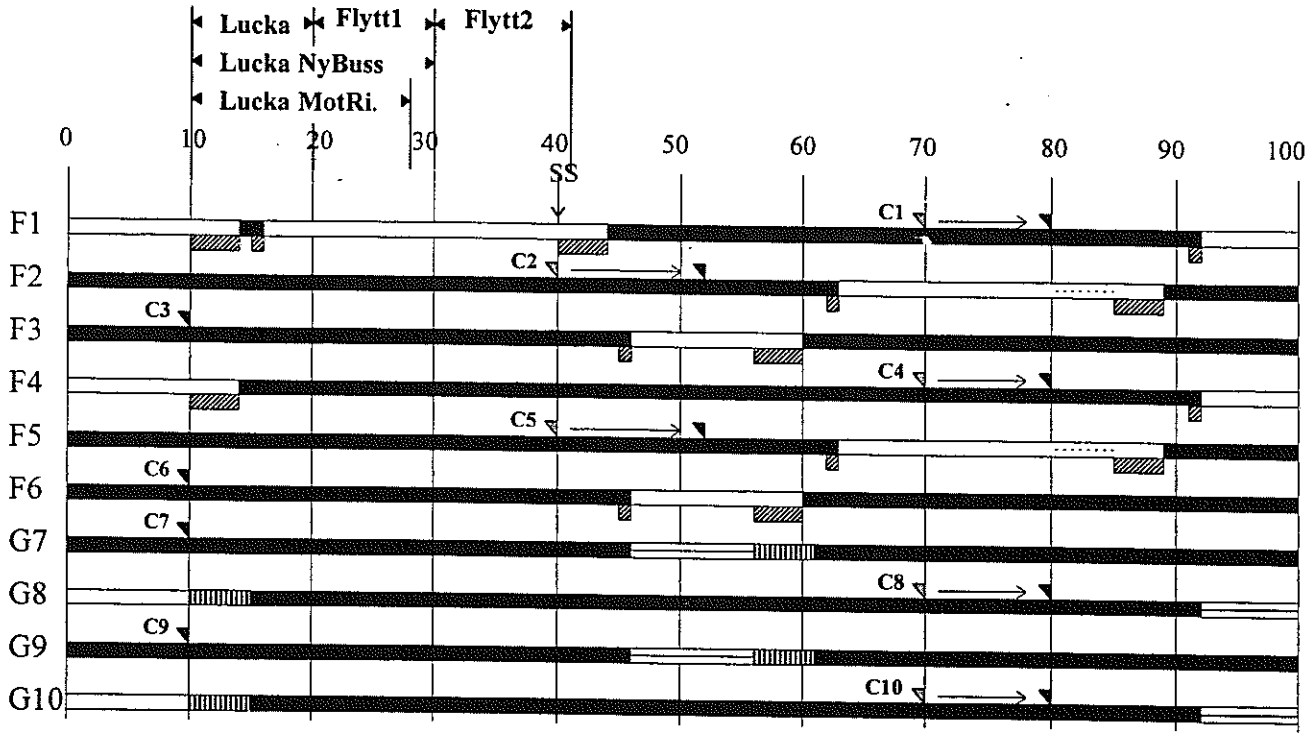


1	Typ av funktion	BF,ÅTS.....	1,2..	**BF2**
6	Förlängande detektor	SGx Mi Fi sek.	DP	**SG1, SG 4 M 0.1 F 0.1 sek.**
16	Lucka	Px;Cx+x sek.-Cx+x sek.	2-3	**P1; C3-C3+5**
	Lucka NyBuss	Px;Cx+x sek.-Cx+x sek.	4-5	**P1; C3-C3+20**
	Lucka Motriktad	Px;Cx+x sek.-Cx+x sek.	6-7	**P1; C3-C3+18**
21	Lucka Flytt1	Px;Cx+x sek.-Cx+x sek.	8-9	**P1; C3+10-C3+20**
	Lucka Flytt2	Px;Cx+x sek.-Cx+x sek.	10-11	**P1; C3+20-C2+1**
22	Flytt1, Starter	Px;Cx+x sek.	12	**P1; C2,C5+7 sek C1,C4,C8,C10+5 sek**
24	Flytt2, Starter	Px;Cx+x sek.	13	**P1; C2,C5+12 sek C1,C4,C8,C10+10 sek**
26	Status; måste grön	SGx	16	**SG1, SG4**
41	För; SG måste grön	SGx	24	**SG1**
43	För; SG förlängas	SGx M/Fx sek.	25/20	**SG1 F 50 sek., SG4 F 50 sek**
47	Sista-Steg	Px;Sx+x sek.	(28,PRP16)	**P1; C2**
48	Till rött vid avslut	SGx M/Fx sek.	20/25	**SG1 F 0 sek., SG4 F 0 sek**

Observera att fråntidsförlängning används och fråntiden nollställs vid logikens avslut vilket medför att signalgrupperna växlar till rött direkt vid avslut oberoende av fientlig start.

Copyright © Stockholms Gatu- och Fastighetskontor

Exempel ÅTS

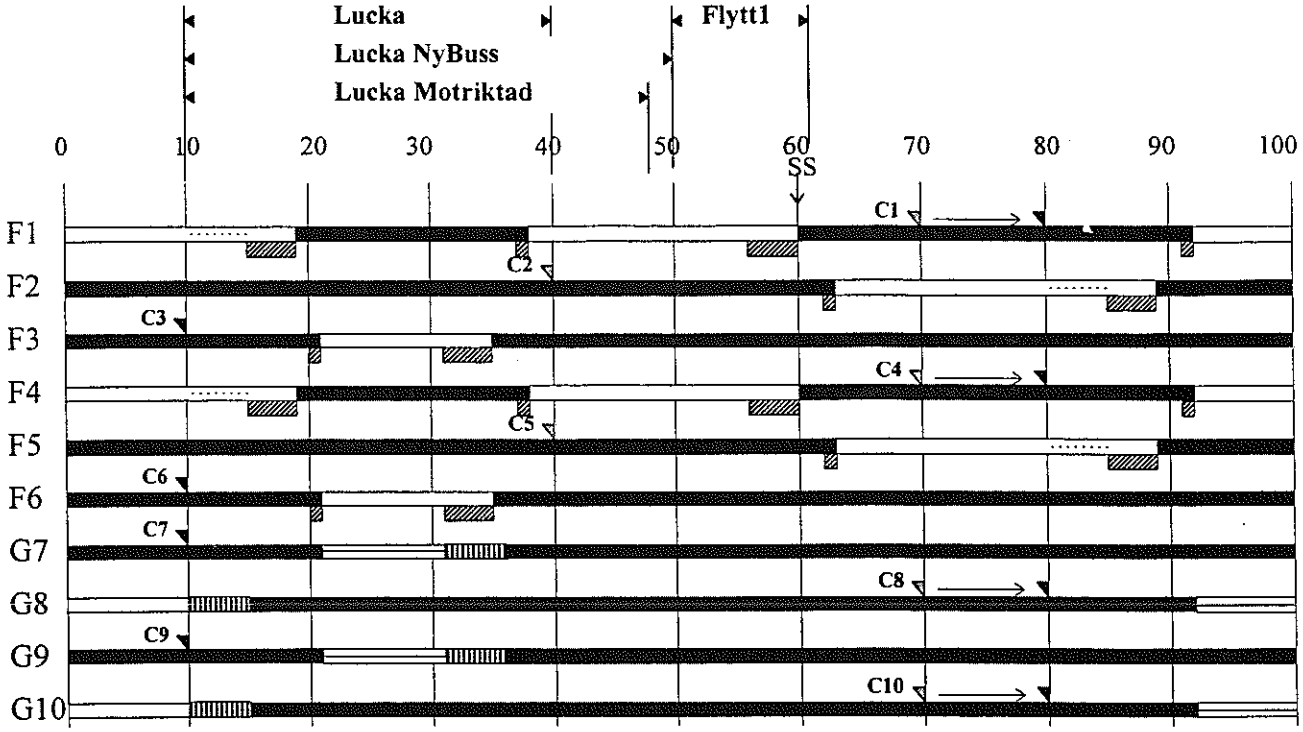


1	Typ av funktion	BF,ÅTS.....	1,2..	**ÅTS**
6	Förlängande detektor	SGx Mi Fi sek.	DP	**SG1 M 0.1 F 0.1 sek.**
16	Lucka	Px;Cx+x sek.-Cx+x sek.	2-3	**P1; C3-C3+5**
	Lucka NyBuss	Px;Cx+x sek.-Cx+x sek.	4-5	**P1; C3-C3+20**
	Lucka Motriktad	Px;Cx+x sek.-Cx+x sek.	6-7	**P1; C3-C3+18**
21	Lucka Flytt1	Px;Cx+x sek.-Cx+x sek.	8-9	**P1; C3+10-C3+20**
	Lucka Flytt2	Px;Cx+x sek.-Cx+x sek.	10-11	**P1; C3+20-C2+1**
22	Flytt1, Starter	Px;Cx+x sek.	12	**P1; C2,C5+7 sek C1,C4,C8,C10+5 sek**
24	Flytt2, Starter	Px;Cx+x sek.	13	**P1; C2,C5+12 sek C1,C4,C8,C10+10 sek**
28	Status; måste röd	SGx	18	**SG1, SG3, SG6, SG7, SG9**
38	Vx2; SG till rött direkt	SGx M/Fx sek.	25/20	**SG3, SG6, SG7, SG9 F 0 sek.**
40	Vx2; SG till grönt	SGx Mx sek.	23	**SG1**
41	För; SG måste grön	SGx	24	**SG1**
43	För; SG förlängas	SGx M/Fx sek.	25/20	**SG1 M 50 sek**
47	Sista-Steg	Px;Sx+x sek.	(28,PRP16)	**P1; C2**
48	Till rött vid avslut	SGx M/Fx sek.	20/25	**SG1 F 0 sek**
49	Till grönt vid avslut	SGx M sek.	23	**SG3, SG6, SG7, SG9 M 0 sek.**

Observera att maxtidsförlängning används och fråntiden nollställs vid logikens avslut vilket medför att signalgrupperna växlar till rött direkt vid avslut. (Maxtiden nollställs automatisk.)

Copyright © Stockholms Gatu- och Fastighetskontor

Exempel EF1

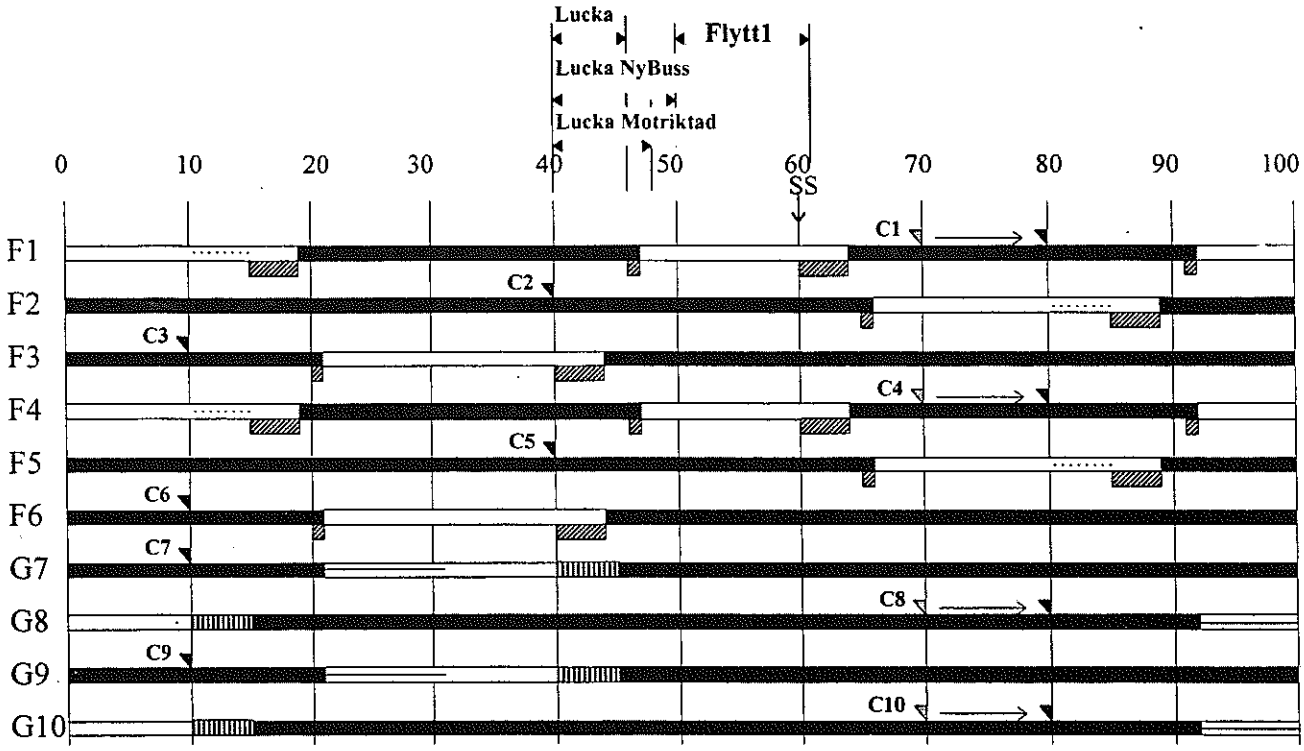


1	Typ av funktion	BF,ÅTS.....	1,2..	**EF1**
6	Förlängande detektor	SGx Mi Fi sek.	DP	**SG1 M 0.1 F 0.1 sek.**
16	Lucka	Px;Cx+x sek.-Cx+x sek.	2-3	**P1; C3-C2**
	Lucka NyBuss	Px;Cx+x sek.-Cx+x sek.	4-5	**P1; C3-C2+10**
	Lucka Motriktad	Px;Cx+x sek.-Cx+x sek.	6-7	**P1; C3-C2+8**
21	Lucka Flytt1	Px;Cx+x sek.-Cx+x sek.	8-9	**P1; C2+10-C2+21**
22	Flytt1, Starter	Px;Cx+x sek.	12	**P1; C2,C5=För C1,C4,C8,C10+10 sek**
27	Status; kan grön	SGx	17	**SG3, SG6, SG7, SG9**
36	Vx2; SG måste grön	SG x	24	**SG3, SG6, SG7, SG9**
39	Vx2; SG till rött garanti	SGx Fx sek.	21	**SG3, SG6, SG7, SG9 F 0 sek.**
40	Vx2; SG till grönt	SGx Mx sek.	23	**SG1, SG 4 M 0 sek.**
41	För; SG måste grön	SGx	24	**SG1**
43	För; SG förlängas	SGx M/Fx sek.	25/20	**SG1 M 50 sek**
47	Sista-Steg	Px;Sx+x sek.	(28,PRP16)	**P1; C2+20**
48	Till rött vid avslut	SGx M/Fx sek.	20/25	**SG1 F 0 sek., SG4 F 0 sek**
49	Till grönt vid avslut	SGx M sek.	23	**SG2 M 0 sek., SG5 M 0 sek.**

Observera att maxtidsförlängning används och fråntiden nollställs vid logikens avslut vilket medför att signalgrupperna växlar till rött direkt vid avslut. (Maxtiden nollställs automatisk.)

Copyright © Stockholms Gatu- och Fastighetskontor

Exempel EF2

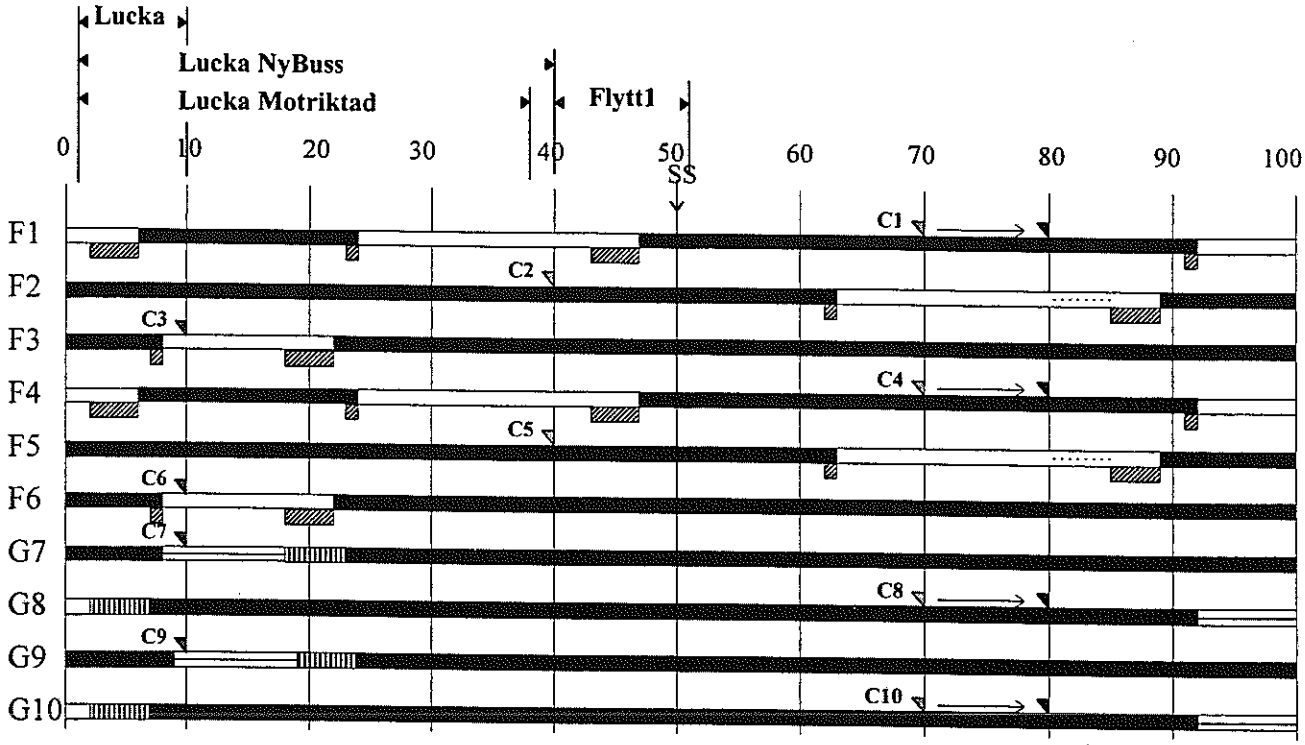


1	Typ av funktion	BF,ÅTS.....	1,2..	**EF2**
6	Förlängande detektor	SGx Mi Fi sek.	DP	**SG1 M 0.1 F 0.1 sek.**
16	Lucka	Px;Cx+x sek.-Cx+x sek.	2-3	**P1; C2-C2+6**
	Lucka NyBuss	Px;Cx+x sek.-Cx+x sek.	4-5	**P1; C2-C2+10**
	Lucka Motriktad	Px;Cx+x sek.-Cx+x sek.	6-7	**P1; C2-C2+8**
21	Lucka Flytt1	Px;Cx+x sek.-Cx+x sek.	8-9	**P1; C2+10-C2+21**
22	Flytt1, Starter	Px;Cx+x sek.	12	**C1,C4,C8,C10+10 sek**
28	Status; måste röd	SGx	18	**SG2, SG3, SG5, SG6, SG7, SG9**
38	Vx2; SG till rött direkt	SGx M/Fx sek.	25/20	**SG2, SG5**
40	Vx2; SG till grönt	SGx Mx sek.	23	**SG1, SG 4**
41	För; SG måste grön	SGx	24	**SG1**
43	För; SG förlängas	SGx M/Fx sek.	25/20	**SG1 M 50 sek**
47	Sista-Steg	Px;Sx+x sek.	(28,PRP16)	**P1; C2+20**
48	Till rött vid avslut	SGx M/Fx sek.	20/25	**SG1 F 0 sek.,SG 4 F 0 sek**
49	Till grönt vid avslut	SGx M sek.	23	**SG2 M 0 sek., SG5 M 0 sek.**

Observera att maxtidsförlängning används och frånttiden nollställs vid logikens avslut vilket medför att signalgrupperna växlar till rött direkt vid avslut. (Maxtiden nollställs automatisk.)

Copyright © Stockholms Gatu- och Fastighetskontor

Exempel DEF = dubbel avkortning med extrafas



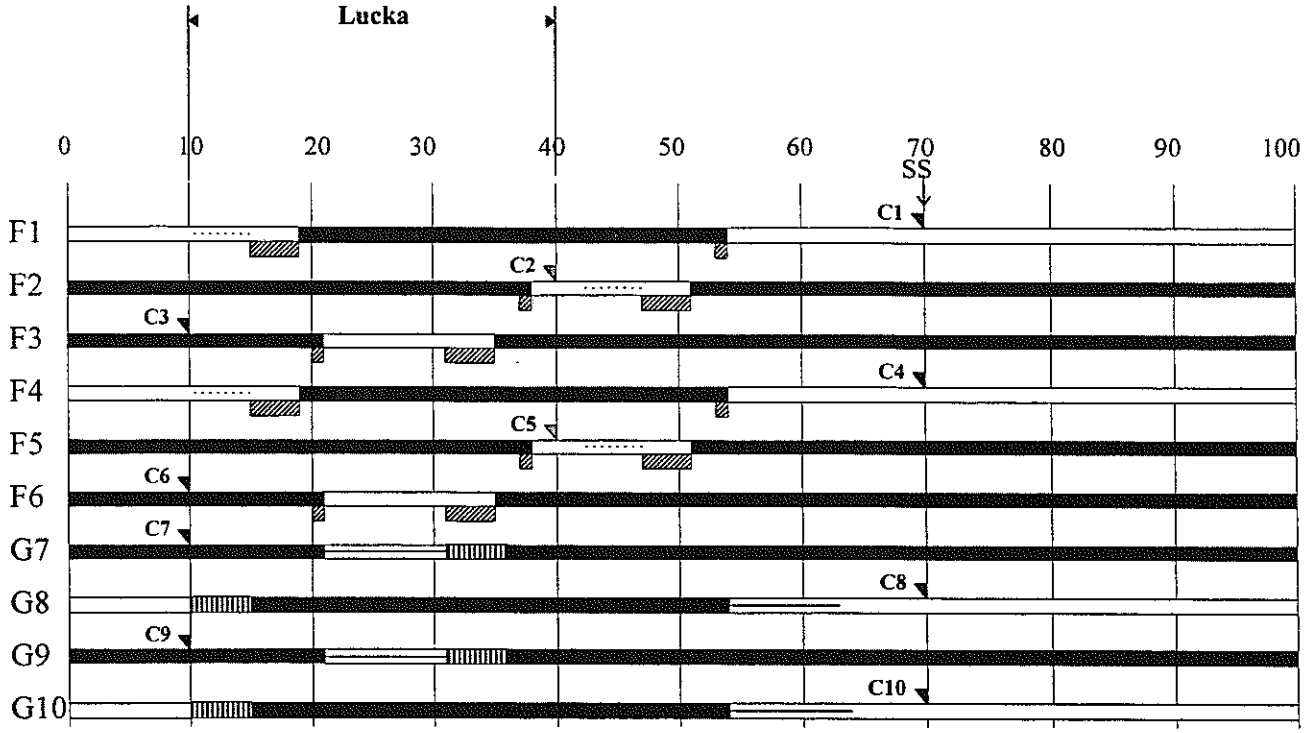
1	Typ av funktion	BF,ÅTS.....	1,2..	**DEF**
6	Förlängande detektor	SGx Mi Fi sek.	DP	**SG1 M 0.1 F 0.1 sek.**
16	Lucka	Px;Cx+x sek.-Cx+x sek.	2-3	**P1; C1+31-C3**
	Lucka NyBuss	Px;Cx+x sek.-Cx+x sek.	4-5	**P1; C1+31-C2**
	Lucka Motriktad	Px;Cx+x sek.-Cx+x sek.	6-7	**P1; C1+31-C3+28**
21	Lucka Flytt1	Px;Cx+x sek.-Cx+x sek.	8-9	**P1; C2-C2+11**
22	Flytt1, Starter	Px;Cx+x sek.	12	**C3,C6,C7,C9,C2,C5=För C1,C4,C8,C10+10**
27	Status; kan grön	SGx	17	**SG1, SG4, SG8, SG10**
31	Vx1; SG måste grön	SGx	24	**SG1, SG4, SG8, SG10**
34	Vx1; SG till rött garanti	SGx Fx sek.	21	**SG1, SG4, SG8, SG10 F 0 sek.**
35	Vx1; SG till grönt	SGx Mx sek.	23	**SG3, SG 6, SG7, SG9 M 0 sek.**
36	Vx2; SG måste grön	SGx	24	**SG3, SG 6, SG7, SG9**
39	Vx2; SG till rött garanti	SGx Fx sek.	21	**SG3, SG 6, SG7, SG9 F 0 sek.**
40	Vx2; SG till grönt	SGx Mx sek.	23	**SG1 M 0 sek., SG 4 M 0 sek.**
41	För; SG måste grön	SGx	24	**SG1**
43	För; SG förlängas	SGx M/Fx sek.	25/20	**SG1 M 50 sek**
47	Sista-Steg	Px;Sx+x sek.	(28,PRP16)	**P1; C2+10**
48	Till rött vid avslut	SGx M/Fx sek.	20/25	**SG1F 0 sek., SG4 F 0 sek**
49	Till grönt vid avslut	SGx M sek.	23	**SG2 M 0 sek., SG5 M 0 sek.**

Kommihåg: Inga luckor får öppnas eller stängas i sekundsteg 0.

Observera att maxtidsförlängning används och fråntiden nollställs vid logikens avslut vilket medför att signalgrupperna växlar till rött direkt vid avslut. (Maxtiden nollställs automatisk.)

Copyright © Stockholms Gatu- och Fastighetskontor

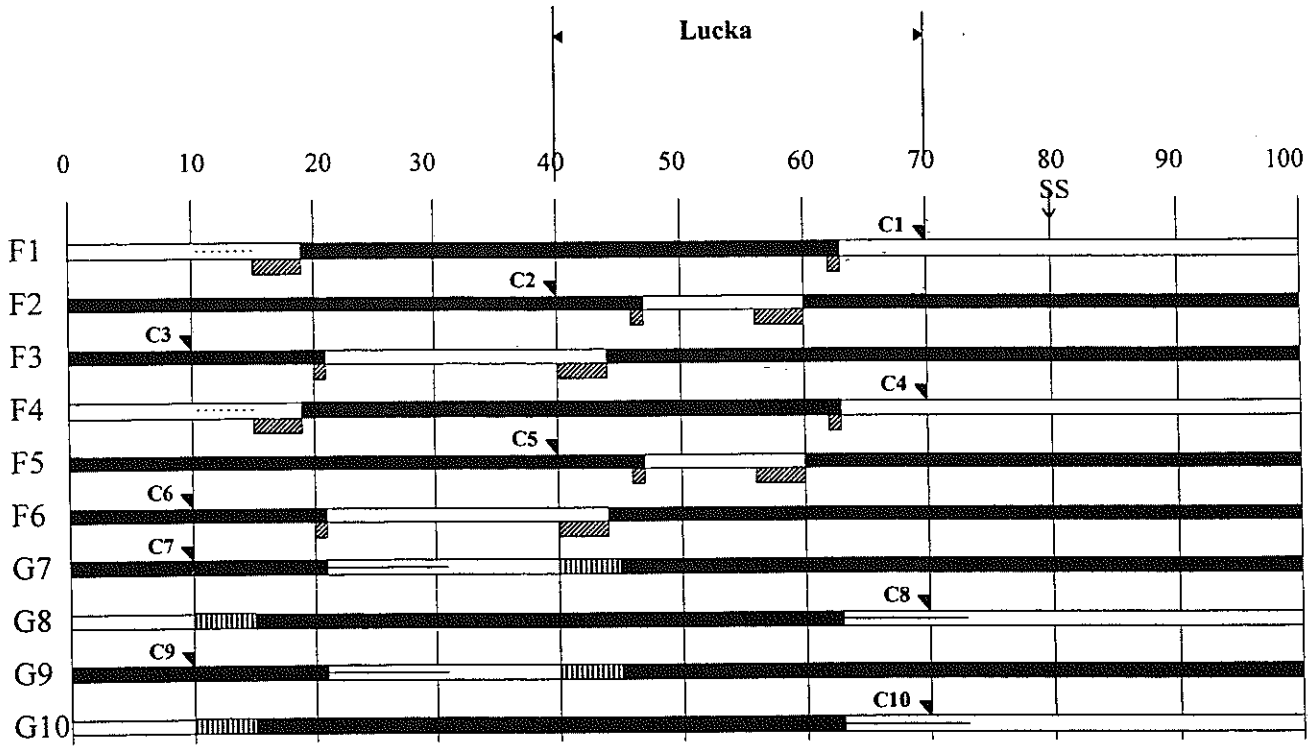
Exempel DAK



1	Typ av funktion	BF,ÅTS.....	1,2..	**DAK**
6	Förlängande detektor	SGx Mi Fi sek.	DP	**SG1 M 0.1 F 0.1 sek.**
16	Lucka	Px;Cx+x sek.-Cx+x sek.	2-3	**P1; C3-C2**
22	Flytt1, Starter	Px;Cx+x sek.	12	**C2,C5=För**
27	Status; kan grön	SGx	17	**SG3, SG6, SG7, SG9**
31	Vx1; SG måste grön	SGx	24	**SG3, SG6, SG7, SG9**
34	Vx1; SG till rött garanti	SGx Fx sek.	21	**SG3, SG6, SG7, SG9**
35	Vx1; SG till grönt	SGx Mx sek.	23	**SG2 M 0 sek., SG 5 M 0 sek.**
36	Vx2; SG måste grön	SGx	24	**SG2, SG 5**
39	Vx2; SG till rött garanti	SGx Fx sek.	21	**SG2 F 5 sek., SG 5 F 5 sek.**
40	Vx2; SG till grönt	SGx Mx sek.	23	**SG1, SG 4, SG8, SG10 M 0 sek.**
47	Sista-Steg	Px;Sx+x sek.	(28,PRP16)	**P1; C1**

Copyright © Stockholms Gatu- och Fastighetskontor

Exempel AK



1	Typ av funktion	BF,ÅTS.....	1,2..	**AK**
6	Förlängande detektor	SGx Mi Fi sek.	DP	**SG1 M 0.1 F 0.1 sek.**
16	Lucka	Px;Cx+x sek.-Cx+x sek.	2-3	**P1; C2-C1**
27	Status; kan grön	SGx	17	**SG2, SG5**
36	Vx2; SG måste grön	SGx	24	**SG2, SG 5**
39	Vx2; SG till rött garanti	SGx Fx sek.	21	**SG2, SG 5**
40	Vx2; SG till grönt	SGx Mx sek.	23	**SG1, SG 3, SG8, SG10**
47	Sista-Steg	Px;Sx+x sek.	(28,PRP16)	**P1; C1+10**

Copyright © Stockholms Gatu- och Fastighetskontor

Blankett för PRIBUSS i ELC-3

Anl. nr:		**Adress:**			
Funktion nr:		**Projektör:**			
Ant. funktioner:		**Beställare**			
Tillfart:		**Datum:**		**Busslinje:**	

1	Typ av funktion	BF,ÅTS..... 1,2..	
2			
3	Aktiv i tidplan	Px 1/27	
4			
5			
6	Förlängande detektor	SGx Mi Fi sek. DP	
7	Räknare	Rx PR	
8	Anmälande detektor	Det x In	
9	Avanmälande detektor	Det x In	
10			
11	Time-Out	x sek. sek	
12	Tillfart	SGx SG	
13	Motsatt räknare	Rx PR	
14			
15			
16	Lucka	Px;Cx+x sek.-Cx+x sek. 2-3	
	Lucka NyBuss	Px;Cx+x sek.-Cx+x sek. 4-5	
	Lucka Motriktad	Px;Cx+x sek.-Cx+x sek. 6-7	
17	Prio.beg. från start	Fx-x sek. 14	
18	Prio.beg. från start	Fx-x sek. 14	
19	Prio.beg. från avslut	Fx-x sek. 15	
20	Prio.beg. från avslut	Fx-x sek. 15	
21	Lucka Flytt1	Px;Cx+x sek.-Cx+x sek. 8-9	
	Lucka Flytt2	Px;Cx+x sek.-Cx+x sek. 10-11	
22	Flytt1, Starter	Px;Cx+x sek. 12	
23	Flytt1, Starter	Px;Cx+x sek. 12	
24	Flytt2, Starter	Px;Cx+x sek. 13	
25	Flytt2, Starter	Px;Cx+x sek. 13	

Anl. nr:		Funktion nr:		
26	Status; måste grön	SGx	16	
27	Status; kan grön	SGx	17	
28	Status; måste röd	SGx	18	
29	Status; kan röd	SGx	19	
30	Status; annan signal	Anl.x, SGx	In	
31	Vx1; SG måste grön	SGx	24	
32				
33	Vx1; SG till rött direkt	SGx M/Fx sek.	25/20	
34	Vx1; SG till rött garanti	SGx Fx sek.	21	
35	Vx1; SG till grönt	SGx Mx sek.	23	
36	Vx2; SG måste grön	SGx	24	
37				
38	Vx2; SG till rött direkt	SGx M/Fx sek.	25/20	
39	Vx2; SG till rött garanti	SGx Fx sek.	21	
40	Vx2; SG till grönt	SGx Mx sek.	23	
41	För; SG måste grön	SGx	24	
42				
43	För; SG förlängas	SGx M/Fx sek.	25/20	
44	För; SG till rött garanti	SGx Fx sek.	21	
45				
46	Ny Time-Out SG grön	x sek.	sek	
47	Sista-Steg	Px;Sx+x sek. (28,PRP16)		
48	Till rött vid avslut	SGx M/Fx sek.	20/25	
49	Till grönt vid avslut	SGx M sek.	23	
50	Förhindra egen funktion	x sek.	sek	

Anmärkningar:

GATU- OCH FASTIGHETS-
KONTORET

Programmeringsanvisningar för PRIPTS. ELC-3 V3.21

av
Tor Thomassen

1996-07-03

Tittel: Programmeringsanvisningar för PRIPTS
Författare: Tor Thomassen
Datum: 96-12-22
Fil: \test\peek\proganv
Version: 1.1

Programmeringsanvisningar för PRIPTS.
ELC-3 V3.21

Detta dukoment beskriver programmeringsmöjligheterna i PRIPTS-logiken som finns i ELC-3 version 3.21. PRIPTS-logiken är baserad på PRIBUSS.

PRIPTS-funktionerna som finns i ELC-3 ägs av Stockholm Gatu- och Fastighetskontor. PRIPTS-funktionerna får ej användas utan skriftlig medgivande från Stockholms Gatu- och Fastighetskontor.

Detta dokument får inte kopieras utan skriftligt medgivande från Stockholms Gatu- och Fastighetskontor.

Dokumentet beskriver hur PRIPTS-logiken skall användas vid "normal" programmering. Med "normal" programmering menas den standard som Signalbyrån på Stockholms Gatu- och Fastighetskontor respektive Trafiktjänsten, Stockholm Entrepenad, har använt vid PRIBUSS-projektering respektive programmering. För en mer detaljerad analys av PRIPTS-funktionerna hänvisas till Peek Traffic User's Description for PRIPTS.

Dokumentet ersätter version 1.0 med utbytesblad.

Referenser:

User's description PRIPTS in ELC-3.	Peek Traffic
Principiellt programmeringsunderlag för PRIBUSS-funktioner i styrapparatens grundprogram.	Stockholm Konsult
PRIBUSS Projekterings- och programmeringsanvisningar.	Stockholms Gatukontor

Copyright © Stockholms Gatu- och Fastighetskontor

Innehåll

Sida

3 Förkortningar och förklarningar.
5 Beskrivning.
7 Programmering i PRP-listorna.
23 Programmering i PGP-listorna.
25 Exempel.
26 Beskrivning av BF.
27 Beskrivning av ÅTS.
28 Programmering av BF i F1.
29 Programmering av BF i F3.
30 Programmering av ÅTS i F1.
31 Programmering av ÅTS i F3.
Bilagor Utskrift av program.
Projekteringsblanketter.

Copyright © Stockholms Gatu- och Fastighetskontor

Förkortningar och förklarningar.

PRIPTS	PRIorization of Public Transport in Signal systems
PRIBUSS	PRIoritering av Bussar i Samordnade Signalsystem
PRP	Funktion
PGP	Definitionslista
RD1	Reduction 1. Växling 1. Första växling vid DAK och DEF.
RD2	Reduction 2. Växling 2. Andra växling vid DAK och DEF eller avkortnig och start till grupper vid vanlig AK, EF och ÅTS
V Ext	Vehicle Extention. Förlägning vid BF, ÅTS, EF och DEF.
ÅTS	ÅterTagenStart.
BF	BussFörlägning.
EF	Extra Fas.
AK	AvKortning.
DAK	DubbelAvKortning.
DEF	Dubbelavkortning med Extra Fas.
TO	TimeOut
AD	Anmälande Detektor
AVD	AvAnmälande Detektor

Copyright © Stockholms Gatu- och Fastighetskontor

Lucka	Fönster i omloppet som bestämmer om funktionen skall tillåtas.
Lucka Ny	Fönster i omloppet som bestämmer om ny buss skall tillåtas.
Lucka Ny Mot	Fönster i omloppet som bestämmer om ny buss i motriktat körfält skall tillåtas.
Sista-Steg	Plats i omloppet där funktionen avslutas oberoende av om buss har avanmält sig eller inte.

Copyright © Stockholms Gatu- och Fastighetskontor

Bekrivning.

PRIPTS-logiken består av två parameterlistor; PRP och PGP. Varje PRP representerar en funktion och består av 54 programmerbara parametrar. Totalt finns det 40 st. PRP-listor.

PGP är en definitionslista där parametrarna används för att beskriva lucka, status, grupper som skall bli gröna etc. Det finns plats för 10 parameter i varje PGP. Totalt finns det 80 st. PGP-listor.

Det finns bara plats för ett tecken i varje parameter i PRP-listan, med undantag för PRP47. För att t.ex beskriva att flera grupper skall bli gröna eller att flera starter som skall flyttas behövs flera parametrar. I PRP programmerar man därför numret till den PGP-lista som innehåller alla de grupper som skall bli gröna eller alla de starter som skall flyttas.

PRP-listorna är generella och används för samtliga funktioner d.v.s. att i en lista kan vilken som helst av funktionerna BF, ÅTS, EF, AK, DAK och DEF programmeras. PRP (kallas hädanefter för **funktionen**) är indelad i 5 moduler, nämligen:

1. Start:	Kontrollerar Lucka, Status, Prioriteringsbegräsningar och yttre villkor.
2. Växling 1:	Kontrollerar grupper, avslutar grupper och startar grupper.
3. Växling 2:	Kontrollerar grupper, avslutar grupper och startar grupper.
4. Förlängning:	Förlänger grupper.
5. Avslut:	Avslutar grupper och startar grupper.

1. Startmodulen startas av puls från AD (anmälande detektor) när luckan för en funktion är öppen. Om status stämmer och det inte finns någon prioriteringsbegränsning fortsätter funktionen med Växling 1. Om status inte är aktivt eller om det finns någon prioriterings-begränsning väntar funktion på att villkoren skall uppfyllas. När alla villkor är uppfyllda fortsätter funktionen. Startmodulen avslutas om luckan stängs eller funktionen avanmäls av puls t.ex. från AVD (avanmälande detektor).

2. Växling 1 används för att ställa grupper till rött samt ge ny start till grupper vid DAK eller DEF. Innan Växling 1 påbörjar sin avkortning av grupper måste vissa angivna grupper vara gröna. Det är lämligt att kontrollera att de grupper som sedan skall ställas till rött verkligen är gröna.

3. Växling 2 används som andra växling under DAK och DEF, men används också när bara en växling skall utföras (t.ex. AK och EF). Innan Växling 2 påbörjar sin avkortning av grupper måste vissa angivna grupper vara gröna.

Copyright © Stockholms Gatu- och Fastighetskontor

4. Förlängning används för att förlänga bussgruppen och allierade grupper under BF, ÅTS, EF och DEF. Det finns två typer av förlängning; maxtidsförlängning och fråntidsförlängning.

5. Avslutmodulen aktiveras av AVD, Sista-Steg eller när TO=0. Grupper som förlängs kan då sättas till rött och efterföljande grupper kan ges start.

Ex. Om man vill programmera en BF, används endast modulerna **Start, Förlängning** och **Avslut**. Alla andra moduler i **funktionen** programmeras med 0. Programmet hoppar över de moduler som inte är programmerade. De grupper som skall förlängas programmeras i definitionslistan

Till dessa 5 moduler tillkommer specialfunktioner för räknaren, flytta starter och prioriteringsbegränsning.

Räknaren aktiveras av anmälande detektor. Räknaren räknas ner av avanmälande detektor. Om TimeOut är lika med noll OCH tillfarten är grön nollställs räknaren. För att spara plats i minnet finns det möjlighet att använda andra funktioners räknare. T.ex. att låta räknaren till en funktion i tillfarten sköta all in- och uträkning.

Funktionen för flyttning av starter bygger på att om prioriteringen avslutas i luckan mellan vissa angivna sekundsteg så skall efterföljande angivna starter flyttas ett antal sekunder. Det finns möjlighet att ange två olika luckor för att aktivera denna funktion. Luckorna kallas **Flytt1** och **Flytt2**. Funktionern fungerar på följande sätt: Om prioriteten avslutas innan luckan för Flytt1 blir aktivt flyttas inga starter alls. Om funktionen avslutas när luckan för Flytt1 är aktiv flyttas de på förhand definerade starterna ett visst angivet antal sekunder och vidare om funktionen avslutas när Flytt1 har blivit passiv och luckan för Flytt2 har blivit aktiv tar Flytt2 över och flyttar de på förhand definerade starterna ett angivet antal sekunder.

Det finns möjlighet att prioriteringsbegränsa varje funktion med en individuell tid i sekunder. Alla funktioner är prioriteringsbegränsade när någon annan funktion är aktiv. Det är normalt bara en funktion som kan vara aktiv. Det finns dock möjlighet att välja bort denna funktion när t.ex är två korsningar är programmerade i en och samma apparat..

Copyright © Stockholms Gatu- och Fastighetskontor

Programmering i PRP-listorna.
(Översättning från projekteringsblanketterna)

PRnPm Där **n** är funktionens nummer och **m** är parameter nummer
PGnPm Där **n** är definitionslistans nummer och **m** är radnummer

Parameter som måste programmeras för att PRIPTS-logiken skall fungera:
COP 19 = 1
COV 23 = Antal PRP
COV 24 = Antal PGP

PRP 1 Function.

Bestämmer om funktionen som skall tillåtas. Måste vara större än noll.

För bättre dokumentation rekommenderas:

1=BF
2=ÅTS
3=EF
4=AK
5=DAK
6=DEF

PRP 2 Activ inpNo.

Här anges den ingång som kan ställa villkor för att funktionen skall tillåtas. Om parametern är noll tillåts funktionen alltid. Om en ingång programmeras, kommer funktionen bara att fungera om ingången är aktiv.

PRP 3 Plan Df/1, 27.

Här anges PGP-nummer för beskrivning av tidplan där funktionen är tillåten samt PGP-adress för Lucka, Lucka Ny, Lucka Ny Mot och Sista-Steg.

Tillåten tidplan.

PGP programmeras på följande sätt: 1-PLNo-PLNo

Ex.: Funktionen är tillåten i program P2, P3 och P4.
PGP programmeras: 1-2-3
1-4-0

Copyright © Stockholms Gatu- och Fastighetskontor

Adress för luckor.

Om funktionen är tillåten i fler tidplaner och luckorna är olika måste en adress programmeras så att funktionen hittar rätt PGP i de olika tidplanerna. **Gäller för parametern i PRP16 samt för parametrarna 21-25.**

PGP programmeras på följande sätt: 27-PLNo-DefAd

I PRP programmeras vilken PGP-lista som skall användas. Men i och med att luckorna kan vara olika i olika tidplaner kan inte funktionen nyttja samma PGP i alla tidplaner. Problemet är löst på så sätt att funktionen känner av vilken tidplan som är aktiv och adderar en siffra till den programmerade siffran för att hitta rätt PGP-lista. **DefAd** motsvarar den siffran som skall adderas till den programmerade siffran i PRP i de olika tidplanerna. **PLNo** motsvarar tidplanerna.

Ex.
27-2-0
27-3-1
27-4-2

Om PRP16(PGP-nummer för Lucka) = 12 kommer funktionen at leta i PGP-nummer **12+0** om program P2 är aktivt och om P4 är aktiv kommer funktionen att leta i PGP-nummer **12+2** efter luckan.

PRP 4 Not used.

PRP 5 Func. outp.

Här anges utgång som aktiveras när funktionen är aktiv. Kan t.ex. användas för ETC med en IOC-utgång.

PRP 6 Request outp.

Här anges utgång som aktiveras när funktionen är aktiv. Här programmeras en software port (201-255) som kan aktivera en detektorlogik som i sin tur förlänger busstillfarten.

PRP 7 in/out cntNo.

Här anges nummer på räknaren som skall påverka funktionen (se PRP 8-9 och 11). Varje funktion har en egen räknare. För att spara plats i minnet programmerar man normalt bara en räknare per tillfart och låter sedan även andra funktioner i tillfarten titta på denna räknare.

Ex. I tillfart F1 finns BF, ÅTS, EF och AK. I BF-funktionen programmeras räknaren och man låter de andra funktionerna använda samma räknare. För ÅTS, EF och AK programmeras då bara PRP 7.

Copyright © Stockholms Gatu- och Fastighetskontor

PRP 8 in cnt inpNo.

Här anges anmälande detektoringång.
Används samma räknare som angetts till annan funktion för tillfarten i PRP7, behövs denna parameter ej programmeras.

PRP 9 out cnt inpNo.

Här anges avanmälande detektoringång.
Används samma räknare som angetts till annan funktion för tillfarten i PRP7, behövs denna parameter ej programmeras.

PRP 10 Not used.

PRP 11 in TimeOut.

Här anges tillfartens TimeOut.
Används samma räknare som angetts till annan funktion för tillfarten i PRP7, behövs denna parameter ej programmeras.

PRP 12 in grpNo.

Här anges busstillfartens signalgruppsnummer så att räknaren vet vilken signalgrupp den tillhör. Räknaren räknar bara när avanmälan sker eller om TO=0 OCH tillfarten är grön. Används samma räknare som angetts till annan funktion för tillfarten i PRP7, behövs denna parameter ej programmeras.

PRP 13 opp. cntNo.

Här anges räknare för motriktad buss. Används bara om buss i motriktat tillfart skall tillåtas förlänga den egna gruppen.

PRP 14 Not used.

PRP 15 Not used.

PRP 16 window def/2-7, 28.

Här anges PGP-nummer för beskrivning av luckor och Sista-Steg.

Copyright © Stockholms Gatu- och Fastighetskontor

Luckor.

Om funktionen är aktiv i flera tidplaner kommer PGP-numret att få ett tillägg. Beroende på vilken tidplan som är aktiv kommer det programmerade PGP-numret att adderas med en siffra (27- PLNo-defAdd) som finns beskrivet i PRP3. Finns det ingen siffra för den aktiva tidplanen kommer det programmerade PGP-numret att användas utan tillägg.

Med **Lucka** menas det område i omloppet där funktionen tillåts starta. **Lucka Ny** beskriver om ny buss i tillfarten skall godkännas för ytterligare förlängning av pågående funktion. **Lucka Ny Mot** beskriver om buss i motriktad eller allierad tillfart skall godkännas för förlängning av pågående funktion. Detta förutsätter att den motriktade tillfarten antingen förlängs av funktionens detektorlogik eller "hänger" på den prioriterade tillfarten.

Lucka:	2-CNo-Time (Öppnar) 3-CNo-Time (Stänger)
Lucka Ny:	4-CNo-Time (Öppnar) 5-CNo-Time (Stänger)
Lucka Ny Mot:	6-CNo-Time (Öppnar) 7-CNo-Time (Stänger)

En lucka öppnas och stängs på vissa angivna tidssteg i omloppet. För att öka flexibiliteten vid äventuella ändringar i tidplanen så programmeras tidsstegen med tider som relateras till valfria NCP-starter enligt:
Lucka: **öppnar:** *Start till viss SG + antal sek.* - Lucka **stänger:** *Start till viss SG + antal sek.*

Ex. En lucka kan vara; C2+10 - C3+0
Start till F2 på sekundsteg 22.
Start till F3 på sekundsteg 40
Luckan är aktiv mellan sekundsteg 32 och 40.

Sista-Steg.

Om funktionen skall tillåtas i flera tidplaner skall också Sista-Steg programmeras i aktuell definitionslista (PGP). Även Sista-Steg anges i förhållande till viss NCP-start.

Sista-Steg: 28-CNo-Time

CNo motsvarar den rad i NCP-tabellen där aktuell start finns och **Time** är antal sekunder efter starten.

Copyright © Stockholms Gatu- och Fastighetskontor

PRP 17 disable PR1 Df/14.

Här anges PGP-nummer för beskrivning av prioriteringsbägrensning av andra funktioner från start.

Defaultvärde är 0 vilket betyder att alla andra funktioner förhindras när funktionen är aktiv. (räknaren är större än noll och lucka och status är aktiv).

Definitionslistan skall innehålla de funktioner som tillåts bli aktiva under en given tid från det att **funktionen** blev aktiv. Funktionerna förhindras när tiden har räknats ned till noll. När **funktionen** avslutas får alla funktioner bli aktiva.

PGP programmeras på följande sätt: 14-PRNo-Time

Ex. 14-10-15
14-11-15

Funktion 10 och 11 tillåts bli aktiv när **funktionen** startar. Alla andra funktioner förhindras från att bli aktiva. 15 sekunder efter att **funktionen** har startat förhindras funktion 10 och funktion 11. När **funktionen** avslutas får alla andra funktioner bli aktiva.

Funktionen skall ändras (hösten '96);

NY FUNKTION:

Samma defaultvärde och funktion som tidigare.

Definitionslistan skall innehålla de funktioner som tillåts bli aktiva efter en angiven tid från det att **funktionen** blivit aktiv. De angivna funktionerna tillåts sedan bli aktiva tiden räknats ner till noll. Alla andra funktioner får bli aktiva när **funktionen** avslutas (även de angivna funktionerna om ovannämda tid är utmätt).

PGP programmeras på följande sätt: 14-PRNo-Time

Ex. 14-10-15
14-11-15

15 sekunder efter att funktionen har startat tillåts funktion 10 och funktion 11 att bli aktiva. Alla funktioner förhindras dock från att bli aktiva tills **funktionen** avslutas.

Copyright © Stockholms Gatu- och Fastighetskontor

PRP 18 disable PR2 Df/14.

Här anges PGP-nummer för beskrivning av prioriteringsbegränsning av ytterligare 10 st. funktioner efter påbörjad prioritering.(Se PRP 17).

PRP 19 enable PR1 Df/15.

Här anges PGP-nummer för beskrivning av prioriteringsbegränsning av andra funktioner efter avslutad prioritet. Defaultvärde är 0 vilket betyder att alla andra funktioner tillåts bli aktiva när funktionen avslutas. Definitionslistan skall innehålla de funktioner som tillåts bli aktiva en angiven tid efter avslutad prioritet. Funktionerna tillåts bli aktiva när tiden har räknats ner till noll, **men funktionen är felprogrammerad; icke angivna funktioner tillåts nu aldrig bli aktiva igen.**

PGP programmeras på följande sätt: 15-PRNo-Time

Ex. 15-10-15
15-11-15
15-2-0
15-3-0

Funktion 10 och funktion 11 tillåts bli aktiva15 sekunder efter avslutad funktion. Funk.2 och 3 tillåts bli aktiva efter avslutad prioritet. Alla andra funktioner tillåts alldrig bli aktiva igen.

För närvarande: **Alla funktioner som inte skall prio.beg måste anges med 0 sekunder.**

PRP 20 enable PR2 Df/15.

Här anges PGP-nummer för beskrivning av prioriteringsbegränsning av ytterligare 10 st. funktioner efter avslutad prioritering. (Se PRP19)

PRP 21 c-p. move winDef/8-11.

Här anges PGP-nummer för beskrivning av luckor för flytt av efterföljande starter.

Flytt1 Om luckan för **Flytt1** är aktiv när funktionen avslutas skall de definerade starterna i PGP-listan flyttas (se PRP22-23).

PGP programmeras på följande sätt: 8-CNo-Time (Öppnar)
9-CNo-Time (Stänger)

Flytt2 Om luckan för **Flytt 2** är aktiv när funktionen avslutas skall de definerade starterna i PGP-listan flyttas (se PRP24-25).

Copyright © Stockholms Gatu- och Fastighetskontor

PGP programmeras på följande sätt: 10-CNo-Time (Öppnar)
11-CNo-Time (Stänger)

Ex. Lucka för Flytt1 skall vara aktiv mellan sekundsteg 25 och 35. Start till F3 ligger på sekundsteg 22.

Luckan beskrivs: C3+3 - C3+13

PGP programmeras på följande sätt: 8-3-3
9-3-13

PRP 22 move 1_1 Df/12, 26.

Här anges PGP-nummer för beskrivning av förhindrande av starter och flyttning av starter om funktionen avslutas när **Flytt1** är aktiv.

Flyttning av starter:

Om funktionen avslutas före luckan för Flytt1 blir aktiv eller om funktionen avslutas efter att luckan för Flytt1 har blivit passiv flyttas inga starter. Flyttning av starter är alltså helt beroende av om det finns en aktiv lucka (se PRP 21). Om det bara finns Flytt1 måste alltså denna lucka stängas efter **Sista-Steg**.

PGP programmeras på följande sätt: 12-CNo-Time

CNo motsvarar den rad i NCP-tabellen där aktuell start finns och **Time** är antalet sekunder som starten skall flyttas.

Flytten gäller alla starter som definerats i PGP från det att luckan öppnas och ett omlopp framöver under förutsättning att funktionen avslutades när luckan var aktiv.

Förhindrande av starter.

Definerade starter förhindras från det att funktionen blir aktiv och ett omlopp framöver. Starter förhindras oberoende av lucka för **Flytt1** och **Flytt2**.

PGP programmeras på följande sätt: 26-CNo-00

CNo motsvarar den rad i NCP-tabellen där aktuell start finns.

PRP 23 move 1_2 Df/12, 26.

Här anges PGP-nummer för beskrivning av förhindrande av ytterligare 10 st. starter och/eller flyttning av starter om funktionen avslutas när **Flytt1** är aktiv. (Se PRP 22)

Copyright © Stockholms Gatu- och Fastighetskontor

PRP 24 move 2_1 Df/12,26.

Här anges PGP-nummer för beskrivning av förhindrande av starter och flyttning av starter om funktionen avslutas när **Flytt2** är aktiv.

Flyttning av starter:

Om funktionen avslutas när luckan för Flytt2 är passiv flyttas inga starter. Luckan måste alltså stängas efter **Sista-Steg**.

PGP programmeras på följande sätt: 13-CNo-Time

CNo motsvarar den rad i NCP-tabellen där aktuell start finns och **Time** är antal sekunder som starten skall flyttas.

Flytten gäller alla definerade starter från det att luckan öppnas och ett omlopp framöver.

Förhindrande av starter.

Starter förhindras från funktionen blir aktiv och ett omlopp framöver. Starter förhindras oberoende av lucka för **Flytt1** och **Flytt2**.

PGP programmeras på följande sätt: 26-CNo-0

CNo motsvarar den rad i NCP-tabellen där aktuell start finns.

PRP 25 move 2_2 Df/13,26.

Här anges PGP-nummer för beskrivning av förhindrande av ytterligare 10 st. starter och/eller flyttning av starter i **Flytt2**. (Se PRP 24).

PRP 26 all green Df/16.

Här anges PGP-nummer för beskrivning av status. Gäller för PRP 26-30. Om flera PRP programmeras måste alla förutsättningar vara sanna.

Här programmerar man de grupper som måste vara gröna för att status skall gälla.

PGP programmeras på följande sätt: 16-GNo-0

GNo motsvarar den signalgrupp som måste vara grön.

Copyright © Stockholms Gatu- och Fastighetskontor

Ex. SG3 och SG4 måste vara gröna för att BF skall aktiveras.
PGP programmeras på följande sätt: 16-3-0
16-4-0

PRP 27 one green Df/17.

Här anges PGP-nummer för beskrivning av status. Gäller för PRP 26-30. Om flera PRP programmeras måste alla förutsättningar vara sanna.

Här programmerar man de grupper som kan vara gröna för att status skall gälla. En grupp måste dock alltid vara grön för att förutsättningen skall vara sann.

PGP programmeras på följande sätt: 17-GNo-0

GNo motsvarar de signalgrupper som kan vara gröna.

Ex. SG3 eller SG4 måste vara grön för att BF skall aktiveras.
PGP programmeras på följande sätt: 17-3-0
17-4-0

PRP 28 all red Df/18.

Här anges PGP-nummer för beskrivning av status. Gäller för PRP 26-30. Om flera PRP programmeras måste alla förutsättningar vara sanna.

Här programmerar man de grupper som måste vara röda för att status skall gälla.

PGP programmeras på följande sätt: 18-GNo-0

GNo motsvarar den signalgrupp som måste vara röd.

Ex. SG3 och SG4 måste vara röda för att ÅTS skall aktiveras.
PGP programmeras på följande sätt: 18-3-0
18-4-0

PRP 29 one red Df/19.

Här anges PGP-nummer för beskrivning av status. Gäller för PRP 26-30. Om flera PRP programmeras måste alla förutsättningar vara sanna.

Här programmerar man de grupper som kan vara röda för att status skall gälla. En grupp måste dock alltid vara röd för att förutsättningen skall vara sann.

Copyright © Stockholms Gatu- och Fastighetskontor

PGP programmeras på följande sätt: 19-GNo-0

GNo motsvarar de signalgrupper som kan vara röda.

Ex. SG3 eller SG4 måste vara röd för att AK skall aktiveras.
PGP programmeras på följande sätt: 19-3-0
19-4-0

PRP 30 status inpNo.

Här anges den ingång som måste vara aktiv för att status skall godkännas. Denna används vid speciella fall där funktionens egen statushantering inte räcker till och man vill använda c.blocksprogrammering för status. Status styrs då med en softwareport (201-255). Kan även kombineras med funktionens egen statushantering, PRP 26-30. Om flera PRP programmeras måste alla förutsättningar vara sanna.

PRP 31 RD1 grn must Df/24.

Här anges PGP-nummer för beskrivning av grupper som måste vara gröna innan Växling 1 kan starta. Funktionen väntar på grupper som har grönbehov och starttillstånd, men fortfarande är röda. Växling 1 används vid DAK och DEF.

PGP programmeras på följande sätt: 24-GNo-0

GNo motsvarar de signalgrupper som måste vara gröna.

Ex. Om F3 och F4 skall avkortas i första växlingen i en DAK är det lämligt att kontrollera att dessa grupper verkligen är gröna innan de avkortas.
PGP programmeras på följande sätt: 24-3-0
24-4-0

PRP 32 Not used.

PRP 33 RD1 red imm Df/20

PGP-nummer för beskrivning av grupper som skall ställas till rött direkt med eller utan trafikstyrd fråntid eller inhiberas med valfri maxtid. Växling 1 används vid DAK och DEF.

PGP programmeras på följande sätt: 20-GNo-PEG
25-GNo-MAX

GNo motsvarar de signalgrupper som skall ställas till rött direkt och **PEG** (Past End Green time) är antal sekunder trafikstyrd fråntid och **MAX** är antal sekunder trafikstyrd maxtid.

Copyright © Stockholms Gatu- och Fastighetskontor

Ex. F7 skall avkortas med 2 sekunder trafikstyrd fråntid och G8 utan fråntid i första växlingen i en DAK.
PGP programmeras på följande sätt: 20-7-2
20-8-0

PRP 34 RD1 red GT Df/21

Här anges PGP-nummer för beskrivning av grupper som skall avkortas till rött efter **Garanti Tid** (GT) med eller utan trafikstyrd fråntid. Alla angivna grupper är beroende av varandra och alla kommer att ligga kvar i grönt tills sista grupp har mätt ut sin mintid och garantitid. Växling 1 används vid DAK och DEF.

PGP programmeras på följande sätt: 21-GNo-PEG

GNo motsvarar de signalgrupper som skall avkortas till rött efter utmätt trafikstyrd GT och **PEG** (Past End Green time) är antal sekunder trafikstyrd fråntid. GT programmeras i GBP11.

Ex. F2 och F3 skall avkortas efter 10 sekunders GT med 2 sekunders fråntid i första växlingen i en DAK.
GBP programmeras på följande sätt: G2BP11=10
G3BP11=10
PGP programmeras på följande sätt: 21-2-2
21-3-2

PRP 35 RD1 to grn Df/23

Här anges PGP-nummer för beskrivning av grupper som skall startas. Växling 1 används vid DAK och DEF.

PGP programmeras på följande sätt: 23-GNo-Max

GNo motsvarar de signalgrupper som skall startas och **MAX** är antal sekunder maxtid. Signalgrupperna startar inte utan grönbehov.

Ex. F1 och F5 skall startas med 0 sekunder maxtid i första växling i en DAK.
PGP programmeras på följande sätt: 23-1-0
23-5-0

PRP 36 RD2 grn must Df/24

PGP-nummer för beskrivning av grupper som måste vara gröna innan Växling 2 kan starta. Funktionen väntar på grupper som har grönbehov och starttillsånd, men fortfarande är röda. Växling 2 används vid ÅTS, EF och AK och som andra växling under DAK och DEF.

Copyright © Stockholms Gatu- och Fastighetskontor

PGP programmeras på följande sätt: 24-GNo-0

GNo motsvarar de signalgrupper som måste vara gröna.

Ex. Om F3 och F4 skall avkortas i AK är det lämligt att kontrollera att dessa grupper verkligen är gröna innan de avkortas till rött.
PGP programmeras på följande sätt: 24-3-0
24-4-0

PRP 37 Not used.

PRP 38 RD2 red imm Df/20.

Här anges PGP-nummer för beskrivning av grupper som skall avkortas till rött direkt med valfri trafikstyrd fråntid eller inhiberas med valfri maxtid. Växling 2 används vid AK, EF och ÅTS och som andra växling under DAK och DEF. Under ÅTS sätts de grupper som har fått startillstånd till rött direkt (red immediatly).

PGP programmeras på följande sätt: 20-GNo-PEG
25-GNo-MAX

GNo motsvarar de signalgrupper som skall ställas till rött direkt och **PEG** (Past End Green time) är antal sekunder trafikstyrd fråntid och **MAX** är antal sekunder trafikstyrd maxtid.

Ex. F7 skall avkortas med 2 sekunder fråntid och G8 utan fråntid i en AK.
PGP programmeras på följande sätt: 20-7-2
20-8-0

PRP 39 RD2 red GT Df/21

Här anges PGP-nummer för beskrivning av grupper som skall avkortas till rött efter trafikstyrd GT med eller utan trafikstyrd fråntid. Alla angivna grupper är beroende av varandra och alla kommer att ligga kvar i grönt tills sista grupp har mätt ut sin mintid och garantitid. Växling 2 används vid ÅTS, EF och AK och som andra växling under DAK och DEF.

PGP programmeras på följande sätt: 21-GNo-PEG

GNo motsvarar de signalgrupper som skall avkortas till rött efter utmätt trafikstyrd GT och **PEG** (Past End Green time) är antal sekunder trafikstyrd fråntid. GT programmeras i GBP11.

Ex. Om F2 och F3 skall avkortas efter 10 sekunders garantitid med 2 sekunder fråntid i en AK.
GBP programmeras på följande sätt: G2BP11=10
G3BP11=10
PGP programmeras på följande sätt: 21-2-2
21-3-2

Copyright © Stockholms Gatu- och Fastighetskontor

PRP 40 RD2 to grn Df/23.

Här anges PGP-nummer för beskrivning av grupper som skall startas. Växling 2 används vid ÅTS, EF och AK och som andra växling under DAK och DEF.

PGP programmeras på följande sätt: 23-GNo-Max

GNo motsvarar de signalgrupper som skall startas och **MAX** är antal sekunder maxtid. Signalgrupperna startar inte utan grönbehov.

Ex. F1 skall startas med 0 sekunder maxtid i en AK.
PGP programmeras på följande sätt: 23-1-0

PRP 41 V Ext grn must Df/24

Här anges PGP-nummer för beskrivning av grupper som måste vara gröna innan grupperna inhiberas för ny maxtidsättning eller fråntidsättning. Funktionen väntar på grupper som har grönbehov och starttillstånd, men fortfarande är röda.

PGP programmeras på följande sätt: 24-GNo-0

GNo motsvarar de signalgrupper som måste vara gröna.

Ex. Om F3 och F4 skall förlängas i EF måste dessa grupper bli gröna innan de inhiberas med en ny maxtid.
PGP programmeras på följande sätt: 24-3-0
24-4-0

PRP 42 Not used.

PRP 43 V Ext (r imm) Df/20, 25

Här anges PGP-nummer för beskrivning av grupper som skall förlängas med maxtid eller fråntid. Fråntid används bara under BF om busstillfarten håller på att mäta ut fråntid när buss anmälar.

PGP programmeras på följande sätt:
Fråntid: 20-GNo-PEG
Maxtid: 25-GNo-MAX

GNo motsvarar de signalgrupper som skall inhiberas för ny tidsättning. **PEG** (Past End Green time) är antal sekunder fråntid och **MAX** är antal sekunder maxtid.

Copyright © Stockholms Gatu- och Fastighetskontor

Ex. F1 skall förlängas med 10 sekunder maxtid i en ÅTS.
PGP programmeras på följande sätt: 25-1-10

PRP 44 V Ext (r GT) Df/21

Används ej i normala fall!

Här anges PGP-nummer för beskrivning av grupper som skall avkortas till rött efter **Garanti Tid** med eller utan trafikstyrd fråntid. Kan användas på signalgrupper som man vill ställa till rött tidigare för att få en smidigare växling. Alla angivna grupper är beroende av varandra och alla kommer att ligga kvar i grönt tills sista grupp har mätt ut sin mintid och garantitid.

PGP programmeras på följande sätt: 21-GNo-PEG

GNo motsvarar de signalgrupper som skall avkortas till rött efter GT och **PEG** (Past End Green time) är antal sekunder trafikstyrd fråntid. GT programmeras i GBP11.

Ex. Om G8 och G9 skall avkortas efter mintid för att få en snabbare växling efter förlängning.
PGP programmeras på följande sätt: 21-8-0
21-9-0

PRP 45 (VE to grn Df/23)

Används ej i normala fall!

Här anges PGP-nummer för beskrivning av grupper som skall startas under förlägning.

PGP programmeras på följande sätt: 23-GNo-Max

GNo motsvarar de signalgrupper som skall startas och **MAX** är antal sekunder maxtid. Signalgruppen startar inte utan grönbehov.

Ex. Om F4 och F5 skall startas med 0 sekunder maxtid under en förlägning.
PGP programmeras på följande sätt: 23-4-0
23-5-0

PRP 46 Special TimeOut.

Här anges uppdatering av TimeOut, i sekunder, när busstillfarten blir grön. Defaultvärdet är 0 och betyder att TO som satts när bussen detekterades gäller. Lämplig använding är vid EF och DEF om anmälande detektor ligger nära hållplats och risken för att TO=0 när tillfarten blir grön, är stor.

Copyright © Stockholms Gatu- och Fastighetskontor

PRP 47 Funct. max CNo-t.

Programmeras normalt i PRP16!

Här anges Sista-Steg. Defaultvärde är 0-0 och betyder att Sista-Steg programmeras i PRP16.

Om funktionen bara används i en tidplan kan Sista-Steg programmeras i denna parameter. Sista-Steg relateras till en start i NCP-tabellen.

PRP programmeras på följande sätt: CNo-Time

CNo motsvarar den rad i NCP-tabellen där aktuell start finns och **Time** är antal sekunder efter starten.

PRP 48 red Df/20 at EOF.

Här anges PGP-nummer för beskrivning av grupper som skall ställas till rött dirkt vid avslut med valfri fråntid eller inhiberas med valfri mxtid.

PGP programmeras på följande sätt: 20-GNo-PEG
25-GNo-MAX

GNo motsvarar de signalgrupper som skall ställas till rött vid avslut och **PEG** (Past End Green time) är antal sekunder trafikstyrd fråntid och MAX är antal sekunder maxtid.

Ex. Bussgruppen F1 skall avslutas med 2 sekunders fråntid efter BF.
PGP programmeras på följande sätt: 20-1-2

PRP 49 grn Df/23 at EOF.

Här anges PGP-nummer för beskrivning av grupper som skall startas efter avslut av funktionen.

PGP programmeras på följande sätt: 23-GNo-Max

GNo motsvarar de signalgrupper som skall startas och **MAX** är antal sekunder maxtid. Signalgrupper startar inte utan grönbehov.

Ex. F4 och F5 skall startas med 0 sekunder maxtid efter en EF.
PGP programmeras på följande sätt: 23-4-0
23-5-0

Copyright © Stockholms Gatu- och Fastighetskontor

PRP 50 disable new func.

Funktionen kan förindras från att återigen bli aktiv en viss angiven tid i sekunder efter avslut.

PRP 51 Not used.

PRP 52 Not used.

PRP 53 Not used.

PRP 54 Not used.

Copyright © Stockholms Gatu- och Fastighetskontor

Programmering i PGP-listorna.

Beskrivning av funktioner i definitionslistan PGP. Det finns 80 PGP med 10 rader i varje.
PG**n**P**m** Där **n** är definitionslistans nummer och **m** är radnummer.

0-0-X	Avslutar listan (End Of Table).
0-1-X	Ingen funktion.
1-PLNo-PLNo	Lista över tillåtna tidplaner.
2-CNo-Time	Start **Lucka** (för att tillåta funktionen).
3-CNo-Time	Stäng **Lucka** (för att tillåta funktionen).
4-CNo-Time	Start **Lucka Ny** (ny buss).
5-CNo-Time	Stäng **Lucka Ny** (ny buss).
6-CNo-Time	Start **Lucka Ny Mot** (motriktad buss).
7-CNo-Time	Stäng **Lucka Ny Mot** (motriktad buss).
8-CNo-Time	Start lucka **Flytt1**.
9-CNo-Time	Stäng lucka **Flytt1**.
10-CNo-Time	Start lucka **Flytt2**.
11-CNo-Time	Stäng lucka **Flytt2**.
12-CNo-Time	Lista över starter som skall flyttas i **Flytt1**.
13-CNo-Time	Lista över starter som skall flyttas i **Flytt2**.
14-PRNo-Time	Lista över funktioner som är tillåtna en angiven tid efter det att funktionen har startat. **Skall ändras till: (hösten '96)** Lista över funktioner som skall förhindras en angiven tid efter det att funktionen har startat.
15-PRNo-Time	Lista över funktioner som skall förhindras en angiven tid efter det att funktionen har avslutats. Alla funktioner som inte anges kommer aldrig att tillåtas starta igen. (Skall ändras våren '97) Om någon funktion skall begränsas måste alla andra funktioner anges med 0 sekunders begränsning.
16-GNo-0	Status. Lista över signalgrupper som måste vara gröna.
17-GNo-0	Status. Lista över signalgrupper där bara en måste vara grön.
18-GNo-0	Status. Lista över signalgrupper som måste vara röda.
19-GNo-0	Status. Lista över signalgrupper där bara en måste vara röd.

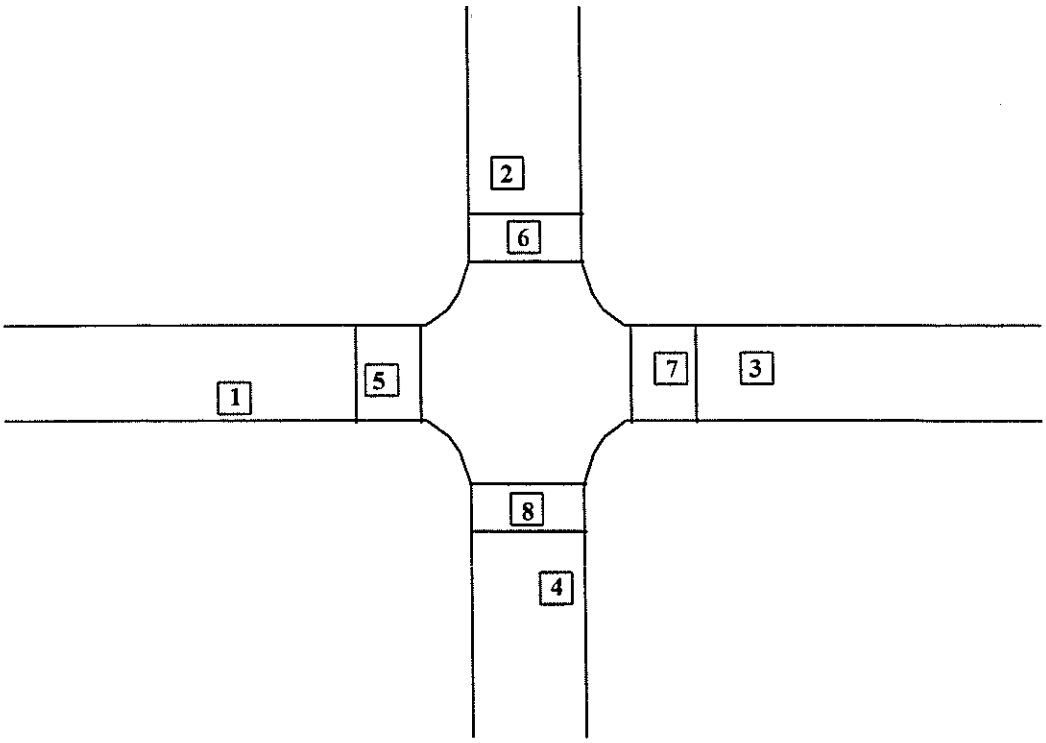
Copyright © Stockholms Gatu- och Fastighetskontor

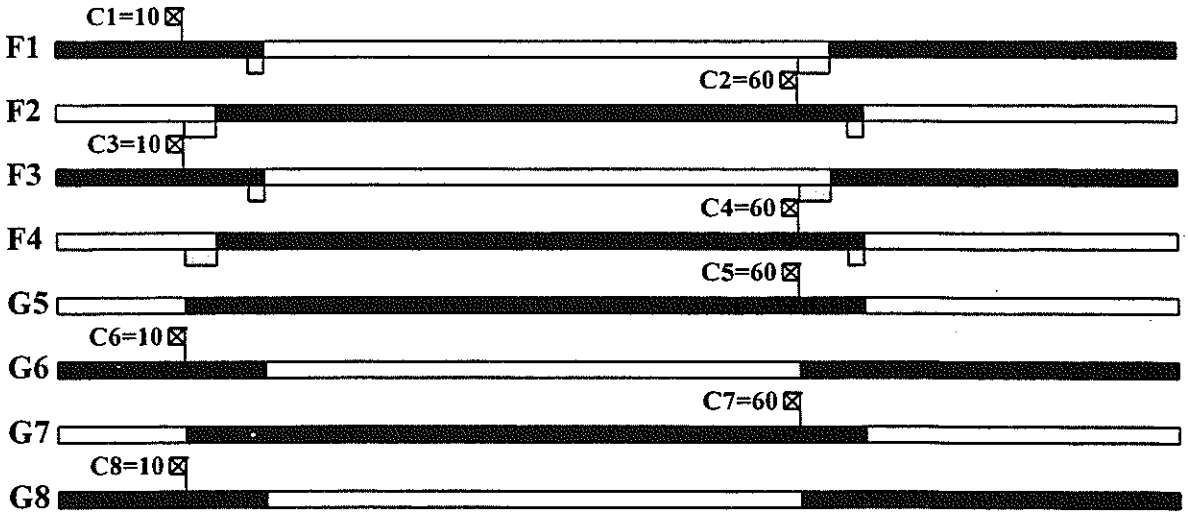
20-GNo-PEG	Lista över signalgrupper som skall sättas till rött med en angiven fråntid. Används också till att förlänga busstillfarten om BF blir aktiv när tillfarten mäter fråntid.
21-GNo-PEG	Lista över signalgrupper som skall sättas till rött efter GT med en angiven fråntid. Alla angivna grupper är beroende av varandra och alla kommer att ligga kvar i grönt tills sista grupp har mätt ut sin mintid och garantitid.
22-0-0	Not Used.
23-GNo-MAX	Lista över signalgrupper som får starttillstånd med en angiven maxtid.
24-GNo-0	Lista över signalgrupper som måste vara gröna innan funktionen fortsätter. (Funktionen väntar tills signalgrupperna är gröna.)
25-GNo-MAX	Lista över signalgrupper som får starttillstånd med en given maxtid. Används för att förlänga busstillfarten.
26-Cno-0	Lista över starter som skall förhindras.
27-PLNo-defAdd	Lista över adresser i olika tidplaner.
28-CNo-Time	Lista överSista-Steg i olika tidplaner.

Copyright © Stockholms Gatu- och Fastighetskontor

Exempel.

Exemplet visar en BF och en ÅTS som skall vara aktiv i P2, P3 och P4.





Signalväxlingsshemat viser **P2, P3** är förskjutet **5 s.** och **P4** är förskjutet **10 sekunder**. Omloppär 80 sekunder i alla program.

Copyright © Stockholms Gatu- och Fastighetskontor

Beskrivning

BF i F1

	P2	P3	P4
TO	10s	10s	10s
Lucka:	C1+40-C2+0 (50-60)	C1+40-C2+0 (55-65)	C1+50-C2+0 (60-70)
Lucka Ny	C1+40-C2+10 (50-70)	C1+40-C2+10 (55-75)	C1+50-C2+10 (60-0)
Lucka Mot	C1+40-C2+5 (50-65)	C1+40-C2+5 (55-70)	C1+50-C2+5 (60-75)
Sista-Steg:	C2+18 (78)	C2+18 (3)	C2+18 (8)
Flytt1	C1+40-C2+10 (50-70)	C1+40-C2+10 (55-75)	C1+40-C2+10 (60-0)
	C2+5 (C2 Flyttas 5s.)	C2+5 (C2 Flyttas 5s.)	C2+5 (C2 Flyttas 5s.)
	C4+5 (C4 Flyttas 5s.)	C4+5 (C4 Flyttas 5s.)	C4+5 (C4 Flyttas 5s.)
	C5+5 (C5 Flyttas 5s.)	C5+5 (C5 Flyttas 5s.)	C5+5 (C5 Flyttas 5s.)
	C7+5 (C7 Flyttas 5s.)	C7+5 (C7 Flyttas 5s.)	C7+5 (C7 Flyttas 5s.)
Flytt2	C2+10-C2+8 (70-78)	C2+10-C2+8 (70-78)	C2+10-C2+8 (70-78)
	C2+5 (C2 Flyttas 10s.)	C2+5 (C2 Flyttas 10s.)	C2+5 (C2 Flyttas 10s.)
	C4+5 (C4 Flyttas 10s.)	C4+5 (C4 Flyttas 10s.)	C4+5 (C4 Flyttas 10s.)
	C5+5 (C5 Flyttas 10s.)	C5+5 (C5 Flyttas 10s.)	C5+5 (C5 Flyttas 10s.)
	C7+5 (C7 Flyttas 10s.)	C7+5 (C7 Flyttas 10s.)	C7+5 (C7 Flyttas 10s.)

BF i F3

TO	20s	20s	20s
Lucka:	C1+40-C2+0 (50-60)	C1+40-C2+0 (55-65)	C1+40-C2+0 (60-70)
Lucka Ny	C1+40-C2+5 (50-65)	C1+40-C2+5 (55-70)	C1+40-C2+5 (60-75)
Lucka Mot	C1+40-C2+10 (50-70)	C1+40-C2+10 (55-75)	C1+40-C2+10 (60-0)
Sista-Steg:	C2+18 (78)	C2+18 (3)	C2+18 (8)
Flytt1	C1+40-C2+10 (50-70)	C1+40-C2+10 (55-75)	C1+40-C2+10 (60-0)
	C2+5 (C2 Flyttas 5s.)	C2+5 (C2 Flyttas 5s.)	C2+5 (C2 Flyttas 5s.)
	C4+5 (C4 Flyttas 5s.)	C4+5 (C4 Flyttas 5s.)	C4+5 (C4 Flyttas 5s.)
	C5+5 (C5 Flyttas 5s.)	C5+5 (C5 Flyttas 5s.)	C5+5 (C5 Flyttas 5s.)
	C7+5 (C7 Flyttas 5s.)	C7+5 (C7 Flyttas 5s.)	C7+5 (C7 Flyttas 5s.)
Flytt2	C2+10-C2+8 (70-78)	C2+10-C2+8 (70-78)	C2+10-C2+8 (70-78)
	C2+5 (C2 Flyttas 10s.)	C2+5 (C2 Flyttas 10s.)	C2+5 (C2 Flyttas 10s.)
	C4+5 (C4 Flyttas 10s.)	C4+5 (C4 Flyttas 10s.)	C4+5 (C4 Flyttas 10s.)
	C5+5 (C5 Flyttas 10s.)	C5+5 (C5 Flyttas 10s.)	C5+5 (C5 Flyttas 10s.)
	C7+5 (C7 Flyttas 10s.)	C7+5 (C7 Flyttas 10s.)	C7+5 (C7 Flyttas 10s.)

Copyright © Stockholms Gatu- och Fastighetskontor

Beskrivning

ÅTS i F1

	P2	P3	P4
TO	10s	10s	10s
Lucka:	C2+0 -C2+5 (60-65)	C2+0-C2+5 (65-70)	C2+5-C2+0 (70-75)
Lucka Ny	C2+0-C2+10 (60-70)	C2+0-C2+10 (65-75)	C2+10-C2+10 (70-0)
Sista-Steg:	C2+18 (78)	C2+18 (3)	C2+18 (8)
Flytt1	C1+40-C2+10 (50-70)	C1+40-C2+10 (55-75)	C1+40-C2+10 (60-0)
	C2+5 (C2 Flyttas 5s.)	C2+5 (C2 Flyttas 5s.)	C2+5 (C2 Flyttas 5s.)
	C4+5 (C4 Flyttas 5s.)	C4+5 (C4 Flyttas 5s.)	C4+5 (C4 Flyttas 5s.)
	C5+5 (C5 Flyttas 5s.)	C5+5 (C5 Flyttas 5s.)	C5+5 (C5 Flyttas 5s.)
	C7+5 (C7 Flyttas 5s.)	C7+5 (C7 Flyttas 5s.)	C7+5 (C7 Flyttas 5s.)
Flytt2	C2+10-C2+8 (70-78)	C2+10-C2+8 (70-78)	C2+10-C2+8 (70-78)
	C2+5 (C2 Flyttas 10s.)	C2+5 (C2 Flyttas 10s.)	C2+5 (C2 Flyttas 10s.)
	C4+5 (C4 Flyttas 10s.)	C4+5 (C4 Flyttas 10s.)	C4+5 (C4 Flyttas 10s.)
	C5+5 (C5 Flyttas 10s.)	C5+5 (C5 Flyttas 10s.)	C5+5 (C5 Flyttas 10s.)
	C7+5 (C7 Flyttas 10s.)	C7+5 (C7 Flyttas 10s.)	C7+5 (C7 Flyttas 10s.)

ÅTS i F3

TO	20s	20s	20s
Lucka:	C2+0-C2+5 (60-65)	C2+0-C2+5 (65-70)	C2+0-C2+5 (70-75)
Lucka Ny	C2+0-C2+5 (60-65)	C2+0-C2+5 (65-70)	C2+0-C2+5 (70-75)
Sista-Steg:	C2+18 (78)	C2+18 (3)	C2+18 (8)
Flytt1	C1+40-C2+10 (50-70)	C1+40-C2+10 (55-75)	C1+40-C2+10 (60-0)
	C2+5 (C2 Flyttas 5s.)	C2+5 (C2 Flyttas 5s.)	C2+5 (C2 Flyttas 5s.)
	C4+5 (C4 Flyttas 5s.)	C4+5 (C4 Flyttas 5s.)	C4+5 (C4 Flyttas 5s.)
	C5+5 (C5 Flyttas 5s.)	C5+5 (C5 Flyttas 5s.)	C5+5 (C5 Flyttas 5s.)
	C7+5 (C7 Flyttas 5s.)	C7+5 (C7 Flyttas 5s.)	C7+5 (C7 Flyttas 5s.)
Flytt2	C2+10-C2+8 (70-78)	C2+10-C2+8 (70-78)	C2+10-C2+8 (70-78)
	C2+5 (C2 Flyttas 10s.)	C2+5 (C2 Flyttas 10s.)	C2+5 (C2 Flyttas 10s.)
	C4+5 (C4 Flyttas 10s.)	C4+5 (C4 Flyttas 10s.)	C4+5 (C4 Flyttas 10s.)
	C5+5 (C5 Flyttas 10s.)	C5+5 (C5 Flyttas 10s.)	C5+5 (C5 Flyttas 10s.)
	C7+5 (C7 Flyttas 10s.)	C7+5 (C7 Flyttas 10s.)	C7+5 (C7 Flyttas 10s.)

Copyright © Stockholms Gatu- och Fastighetskontor

Tittel: Programmeringsanvisningar för PRIPTS
Författare: Tor Thomassen
Datum: 96-12-22
Fil: \test\peek\proganv
Version: 1.1

Programmering

BF i F1.	PR1P		PGP		
1	Function	= 1			
3	Plan Df1	= 1	PG1P1=1-2-3		
			PG1P2=1-4-0		
			PG1P3=27-2-0		
			PG1P4=27-3-1		
			PG1P5=27-4-2		
5	Func. outp.	= 13			
6	Request outp.	= 201			
7	in/out cntNo	= 1			
8	in cnt inpNo	= 31			
9	out cnt inpNo	= 32			
11	in TimeOut	= 10			
12	in grNo	= 1			
13	opp. cntNo	= 2			
16	Window def /2-7	= 2	PG2P1=2-1-40	PG3P1=2-1-40	PG4P1=2-1-40
			PG2P2=3-2-0	PG3P2=3-2-0	PG4P2=3-2-0
			PG2P3=4-1-40	PG3P3=4-1-40	PG4P3=4-1-40
			PG2P4=5-2-10	PG3P4=5-2-10	PG4P4=5-2-10
			PG2P5=6-1-40	PG3P5=6-1-40	PG4P5=6-1-40
			PG2P6=7-2-5	PG3P6=7-2-5	PG4P6=7-2-5
			PG2P7=28-2-18	PG3P7=28-2-18	PG4P7=28-2-18
19	Enable PR1 Df/14	= 2	PG2P8=14-3-20		
			PG2P9=14-4-20		
21	C-p. mov winDef/8-11	= 5	PG5P1=8-1-40	PG6P1=8-1-40	PG7P1=8-1-40
			PG5P2=9-2-10	PG6P2=9-2-10	PG7P2=9-2-10
			PG5P3=10-2-10	PG6P3=10-2-10	PG7P3=10-2-10
			PG5P4=11-2-18	PG6P4=11-2-18	PG7P4=11-2-18
22	Move1_1 Df/12	= 1	PG1P6=12-1-5	PG1P6=12-1-5	PG1P6=12-1-5
			PG1P7=12-3-5	PG1P7=12-3-5	PG1P7=12-3-5
			PG1P8=12-6-5	PG1P8=12-6-5	PG1P8=12-6-5
			PG1P9=12-7-5	PG1P9=12-7-5	PG1P9=12-7-5
24	Move2_1 Df/12	= 5	PG5P5=13-1-10	PG5P5=13-1-10	PG5P5=13-1-10
			PG5P6=13-3-10	PG5P6=13-3-10	PG5P6=13-3-10
			PG5P7=13-6-10	PG5P7=13-6-10	PG5P7=13-6-10
			PG5P8=13-8-10	PG5P8=13-8-10	PG5P8=13-8-10
26	All green Df/16	= 5	PG5P9=16-1-0		
			PG5P10=16-3-0		
41	VE grn must Df/24	= 6	PG6P9=24-1-0		
			PG6P10=24-3-0		
43	V Ext. (r imm) Df/20,25	= 7	PG7P9=25-1-50		
			PG7P10=25-3-50		

Copyright © Stockholms Gatu- och Fastighetskontor

Programmering

BF i F3.	PR2P		PGP		
1	Function	= 1			
3	Plan Df1	= 1	PG1P1=1-2-3		
			PG1P2=1-4-0		
			PG1P3=27-2-0		
			PG1P4=27-3-1		
			PG1P5=27-4-2		
5	Func. outp.	= 14			
6	Request outp.	= 202			
7	in/out cntNo	= 2			
8	in cnt inpNo	= 33			
9	out cnt inpNo	= 34			
11	in TimeOut	= 20			
12	in grNo	= 3			
13	opp. cntNo	= 1			
16	Window def /2-7	= 8	PG8P1=2-1-40	PG9P1=2-1-40	PG10P1=2-1-40
			PG8P2=3-2-0	PG9P2=3-2-0	PG10P2=3-2-0
			PG8P3=4-1-40	PG9P3=4-1-40	PG10P3=4-1-40
			PG8P4=5-2-5	PG9P4=5-2-5	PG10P4=5-2-5
			PG8P5=6-1-40	PG9P5=6-1-40	PG10P5=6-1-40
			PG8P6=7-2-10	PG9P6=7-2-10	PG10P6=7-2-10
			PG8P7=28-2-18	PG9P7=28-2-18	PG10P7=28-2-18
19	Enable PR1 Df/14	= 2	PG2P8=14-3-20		
			PG2P9=14-4-20		
21	C-p. mov winDef/8-11	= 5	PG5P1=8-1-40	PG6P1=8-1-40	PG7P1=8-1-40
			PG5P2=9-2-10	PG6P2=9-2-10	PG7P2=9-2-10
			PG5P3=10-2-10	PG6P3=10-2-10	PG7P3=10-2-10
			PG5P4=11-2-18	PG6P4=11-2-18	PG7P4=11-2-18
22	Move1_1 Df/12	= 1	PG1P6=12-1-5	PG1P6=12-1-5	PG1P6=12-1-5
			PG1P7=12-3-5	PG1P7=12-3-5	PG1P7=12-3-5
			PG1P8=12-6-5	PG1P8=12-6-5	PG1P8=12-6-5
			PG1P9=12-7-5	PG1P9=12-7-5	PG1P9=12-7-5
24	Move2_1 Df/12	= 5	PG5P5=13-1-10	PG5P5=13-1-10	PG5P5=13-1-10
			PG5P6=13-3-10	PG5P6=13-3-10	PG5P6=13-3-10
			PG5P7=13-6-10	PG5P7=13-6-10	PG5P7=13-6-10
			PG5P8=13-8-10	PG5P8=13-8-10	PG5P8=13-8-10
26	All green Df/16	= 5	PG5P9=16-1-0		
			PG5P10=16-3-0		
41	VE grn must Df/24	= 6	PG6P9=24-1-0		
			PG6P10=24-3-0		
43	V Ext. (r imm) Df/20,25	= 7	PG7P9=25-1-50		
			PG7P10=25-3-50		

Copyright © Stockholms Gatu- och Fastighetskontor

Tittel: Programmeringsanvisningar för PRIPTS
Författare: Tor Thomassen
Datum: 96-12-22
Fil: \test\peek\proganv
Version: 1.1

Programmering

ÅTS i F1. PR3P

			PGP		
1	Function	= 2			
3	Plan Df1	= 1	PG1P1=1-2-3		
			PG1P2=1-4-0		
			PG1P3=27-2-0		
			PG1P4=27-3-1		
			PG1P5=27-4-2		
5	Func. outp.	= 15			
6	Request outp.	= 201			
7	in/out cntNo	= 1			
16	Window def /2-7	= 11	PG11P1=2-2-0	PG12P1=2-2-0	PG13P1=2-2-0
			PG11P2=3-2-5	PG12P2=3-2-5	PG13P2=3-2-5
			PG11P3=4-2-0	PG12P3=4-2-0	PG13P3=4-2-0
			PG11P4=5-2-10	PG12P4=5-2-10	PG13P4=5-2-10
			PG11P5=28-2-18	PG12P5=28-2-18	PG13P5=28-2-18
21	C-p. mov winDef/8-11	= 5	PG5P1=8-1-40	PG6P1=8-1-40	PG7P1=8-1-40
			PG5P2=9-2-10	PG6P2=9-2-10	PG7P2=9-2-10
			PG5P3=10-2-10	PG6P3=10-2-10	PG7P3=10-2-10
			PG5P4=11-2-18	PG6P4=11-2-18	PG7P4=11-2-18
22	Move1_1 Df/12	= 1	PG1P6=12-1-5	PG1P6=12-1-5	PG1P6=12-1-5
			PG1P7=12-3-5	PG1P7=12-3-5	PG1P7=12-3-5
			PG1P8=12-6-5	PG1P8=12-6-5	PG1P8=12-6-5
			PG1P9=12-7-5	PG1P9=12-7-5	PG1P9=12-7-5
24	Move2_1 Df/12	= 5	PG5P5=13-1-10	PG5P5=13-1-10	PG5P5=13-1-10
			PG5P6=13-3-10	PG5P6=13-3-10	PG5P6=13-3-10
			PG5P7=13-6-10	PG5P7=13-6-10	PG5P7=13-6-10
			PG5P8=13-8-10	PG5P8=13-8-10	PG5P8=13-8-10
28	All red Df/18	= 11	PG11P6=18-1-0		
			PG11P7=18-2-0		
			PG11P8=18-4-0		
			PG11P9=18-5-0		
			PG11P10=18-7-0		
38	RD2 red imm. Df/20	= 12	PG12P6=20-2-0		
			PG12P7=20-4-0		
			PG12P8=20-5-0		
			PG12P9=20-7-0		
36	RD2 to grn Df/24	= 12	PG12P10=24-1-0		
41	VE grn must Df/24	= 12	PG12P10=24-1-0		
43	V Ext. (r imm) Df/20,25	= 8	PG8P8=25-1-50		
46	Special TimeOut	= 10			
48	Red Df/20 at EOF	= 8	PG8P9=20-1-0		
49	grn Df/23 at EOF	= 13	PG13P6=23-2-0		
			PG13P7=23-4-0		
			PG13P8=23-5-0		
			PG13P9=23-7-0		

Copyright © Stockholms Gatu- och Fastighetskontor

Tittel: Programmeringsanvisningar för PRIPTS
Författare: Tor Thomassen
Datum: 96-12-22
Fil: \test\peek\proganv
Version: 1.1

Programmering

ÅTS i F3. PR3P PGP

1	Function	= 2			
3	Plan Df1	= 1	PG1P1=1-2-3		
			PG1P2=1-4-0		
			PG1P3=27-2-0		
			PG1P4=27-3-1		
			PG1P5=27-4-2		
5	Func. outp.	= 16			
6	Request outp.	= 202			
7	in/out cntNo	= 2			
16	Window def /2-7	= 14	PG14P1=2-2-0	PG15P1=2-2-0	PG16P1=2-2-0
			PG14P2=3-2-5	PG15P2=3-2-5	PG16P2=3-2-5
			PG14P3=4-2-0	PG15P3=4-2-0	PG16P3=4-2-0
			PG14P4=5-2-5	PG15P4=5-2-5	PG16P4=5-2-5
			PG14P5=28-2-18	PG15P5=28-2-18	PG16P5=28-2-18
21	C-p. mov winDef/8-11	= 5	PG5P1=8-1-40	PG6P1=8-1-40	PG7P1=8-1-40
			PG5P2=9-2-10	PG6P2=9-2-10	PG7P2=9-2-10
			PG5P3=10-2-10	PG6P3=10-2-10	PG7P3=10-2-10
			PG5P4=11-2-18	PG6P4=11-2-18	PG7P4=11-2-18
22	Move1_1 Df/12	= 1	PG1P6=12-1-5	PG1P6=12-1-5	PG1P6=12-1-5
			PG1P7=12-3-5	PG1P7=12-3-5	PG1P7=12-3-5
			PG1P8=12-6-5	PG1P8=12-6-5	PG1P8=12-6-5
			PG1P9=12-7-5	PG1P9=12-7-5	PG1P9=12-7-5
24	Move2_1 Df/12	= 5	PG5P5=13-1-10	PG5P5=13-1-10	PG5P5=13-1-10
			PG5P6=13-3-10	PG5P6=13-3-10	PG5P6=13-3-10
			PG5P7=13-6-10	PG5P7=13-6-10	PG5P7=13-6-10
			PG5P8=13-8-10	PG5P8=13-8-10	PG5P8=13-8-10
28	All red Df/18	= 14	PG14P6=18-3-0		
			PG14P7=18-2-0		
			PG14P8=18-4-0		
			PG14P9=18-5-0		
			PG14P10=18-7-0		
38	RD2 red imm. Df/20	= 12	PG12P6=20-2-0		
			PG12P7=20-4-0		
			PG12P8=20-5-0		
			PG12P9=20-7-0		
36	RD2 to grn Df/24	= 15	PG12P10=24-3-0		
41	VE grn must Df/24	= 15	PG12P10=24-1-0		
43	V Ext. (r imm) Df/20,25	= 9	PG8P8=25-3-50		
48	Red Df/20 at EOF	= 9	PG8P9=20-3-0		
49	grn Df/23 at EOF	= 13	PG13P6=23-2-0		
			PG13P7=23-4-0		
			PG13P8=23-5-0		
			PG13P9=23-7-0		

Copyright © Stockholms Gatu- och Fastighetskontor

FUNKTION

			PR1	PR2	PR3	PR4
			BF-F1	BF-F3	ÅTS-F1	ÅTS-F3
1	function	=	1	1	2	2
2	active inpNo	=	0	0	0	0
3	Plan Df/1	=	1	1	1	1
4	Not used	=	0	0	0	0
5	func. outp.	=	13	14	15	16
6	request outp	=	201	202	201	202
7	in/out cntNo	=	1	2	1	2
8	in cnt inpNo	=	31	33	0	0
9	out cnt inpNo	=	32	34	0	0
10	Not used	=	0-0	0-0	0-0	0-0
11	in timeout	=	10	20	0	0
12	in grNo	=	1	3	0	0
13	opp. cntNo	=	2	1	0	0
14	Not used	=	0	0	0	0
15	Not used	=	0	0	0	0
16	window def/2-7	=	2	8	11	14
17	disable PR1 Df/14	=	0	0	0	0
18	disable PR2 Df/14	=	0	0	0	0
19	enable PR1 Df/15	=	2	2	0	0
20	enable PR2 Df/15	=	0	0	0	0
21	c-p. mov winDef/8-11	=	5	5	5	5
22	move1_1 Df/12	=	1	1	1	1
23	move1_2 Df/12	=	0	0	0	0
24	move2_1 Df/13	=	5	5	5	5
25	move2_2 Df/13	=	0	0	0	0
26	all green Df/16	=	5	5	0	0
27	one green Df/17	=	0	0	0	0
28	all red Df/18	=	0	0	11	11
29	one red Def/19	=	0	0	0	0
30	status inpNo	=	0	0	0	0
31	RD1 grn must Df/24	=	0	0	0	0
32	Not used	=	0	0	0	0
33	RD1 red imm. Df/20	=	0	0	0	0
34	RD1 red GT Df/21	=	0	0	0	0
35	RD1 to grn Df/23	=	0	0	0	0
36	RD2 grn must Df/24	=	0	0	0	0
37	Not used	=	0	0	0	0
38	RD2 red imm. Df/20	=	0	0	12	12
39	RD2 red GT Df/21	=	0	0	0	0
40	RD2 to grn Df/23	=	0	0	12	15
41	VE grn must Df/24	=	6	6	12	15
42	Not used	=	0	0	0	0
43	V Ext. (r imm) Df/20	=	7	7	8	9
44	V Ext. (r GT) Df/21	=	0	0	0	0
45	(VE togrn Df/23)	=	0	0	0	0
46	special timeout	=	0	0	10	0
47	funct. max cNo-t	=	0-0	0-0	0-0	0-0
48	red Df/20 at EOF	=	0	0	8	9
49	grn Df/23 at EOF	=	8	0	13	13
50	disable new func.	=	0	0	0	0
51	Not used	=	0	0	0	0
52	Not used	=	0	0	0	0
53	Not used	=	0	0	0	0
54	Not used	=	0	0	0	0

PG

			PG1	PG2	PG3	PG4	PG5	PG6
1	dType-def1-def2	=	1-2-3	2-1-40	2-1-40	2-1-40	8-1-40	8-1-40
2	dType-def1-def2	=	1-4-0	3-2-0	3-2-0	3-2-0	9-2-10	9-2-10
3	dType-def1-def2	=	27-2-0	4-1-40	4-1-40	4-1-40	10-2-10	10-2-10
4	dType-def1-def2	=	27-3-1	5-2-10	5-2-10	5-2-10	11-2-18	11-2-18
5	dType-def1-def2	=	27-4-2	6-1-40	6-1-40	6-1-40	13-1-10	13-1-10
6	dType-def1-def2	=	12-1-5	7-2-5	7-2-5	7-2-5	13-3-10	13-3-10
7	dType-def1-def2	=	12-3-5	28-2-18	28-2-18	28-2-18	13-6-10	13-6-10
8	dType-def1-def2	=	12-6-5	14-3-20	0-0-0	0-0-0	13-8-10	13-8-10
9	dType-def1-def2	=	12-8-5	14-4-20	0-0-0	0-0-0	16-1-0	24-1-0
10	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	16-3-0	24-3-0
		=	PG7	PG8	PG9	PG10	PG11	PG12
1	dType-def1-def2	=	8-1-40	2-1-40	2-1-40	2-1-40	2-2-0	2-2-0
2	dType-def1-def2	=	9-2-10	3-2-0	3-2-0	3-2-0	3-2-5	3-2-5
3	dType-def1-def2	=	10-2-10	4-1-40	4-1-40	4-1-40	4-2-0	4-2-0
4	dType-def1-def2	=	11-2-18	5-2-5	5-2-5	5-2-5	5-2-10	5-2-10
5	dType-def1-def2	=	13-1-10	6-1-40	6-1-40	6-1-40	28-2-18	28-2-18
6	dType-def1-def2	=	13-3-10	7-2-10	7-2-10	7-2-10	18-1-0	20-2-0
7	dType-def1-def2	=	13-6-10	28-2-18	28-2-18	28-2-18	18-2-0	20-4-0
8	dType-def1-def2	=	13-8-10	25-1-50	0-0-0	0-0-0	18-4-0	20-5-0
9	dType-def1-def2	=	25-1-50	20-1-0	0-0-0	0-0-0	18-5-0	20-7-0
10	dType-def1-def2	=	25-3-50	0-0-0	0-0-0	0-0-0	18-7-0	24-1-0
			PG13	PG14	PG15	PG16	PG17	PG18
1	dType-def1-def2	=	2-2-0	2-2-0	2-2-0	2-2-0	0-0-0	0-0-0
2	dType-def1-def2	=	3-2-5	3-2-5	3-2-5	3-2-5	0-0-0	0-0-0
3	dType-def1-def2	=	4-2-0	4-2-0	4-2-0	4-2-0	0-0-0	0-0-0
4	dType-def1-def2	=	5-2-10	5-2-10	5-2-10	5-2-10	0-0-0	0-0-0
5	dType-def1-def2	=	28-2-18	28-2-18	28-2-18	28-2-18	0-0-0	0-0-0
6	dType-def1-def2	=	23-2-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
7	dType-def1-def2	=	23-4-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
8	dType-def1-def2	=	23-5-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
9	dType-def1-def2	=	23-7-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
10	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
			PG19	PG20	PG21	PG22	PG23	PG24
1	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
2	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
3	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
4	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
5	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
6	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
7	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
8	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
9	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
10	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
			PG25	PG26	PG27	PG28	PG29	PG30
1	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
2	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
3	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
4	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
5	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
6	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
7	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
8	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
9	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0
10	dType-def1-def2	=	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0	0-0-0

Anl.Nr:	**01**	**Address:**	**Testvägen-Provgatan**		
Funktion nr:	**1**	**Projektör:**	**Olle Proj**		
Ant. funktioner:	**4**	**Beställare**	**Komunen**		
Tillfart:	**F1**	**Datum:**	**1996-06-27**	**Busslinje:**	**Alla**

1	Typ av funktion	BF,ÅTS.....	**BF**
2			
3	Aktiv i tidplan	Px	**P2, P3, P4**
4			
5			
6	Förlängande detektor	SGx Mx,Fx sek.	**SG1 M=0.1**
7	Räknare	Rx	**R1**
8	Anmälande detektor	Det x	**BD1**
9	Avanmälande detektor	Det x	**BD2**
10			
11	Time-Out	x sek.	**10 sek**
12	Tillfart	SGx	**F1**
13	Motsatt räknare	Rx	**R2**
14			
15			
16	Lucka	Px;Sx+x sek.	**P2; C1+40-C2+0** **P3; C1+40-C2+0** **P4; C1+40-C2+0**
	Lucka NyBuss	Px;Sx+x sek.	**P2; C1+40-C2+10** **P3; C1+40-C2+10** **P4; C1+40-C2+10**
	Lucka Motriktad	Px;Sx+x sek.	**P2; C1+40-C2+5** **P3; C1+40-C2+5** **P4; C1+40-C2+5**
17	Prio.beg. från start	Fx-x sek.	
18	Prio.beg. från start	Fx-x sek.	
19	Prio.beg. från avslut	Fx-x sek.	**F3-20, F4-20**
20	Prio.beg. från avslut	Fx-x sek.	
21	Lucka Flytt1	Px;Sx+x sek.	**P2; C1+40-C2+10** **P3; C1+40-C2+10** **P4; C1+40-C2+10**
	Lucka Flytt2	Px;Sx+x sek.	**P2; C2+10-C2+18** **P3; C2+10-C2+18** **P4; C2+10-C2+18**
22	Flytt1, Starter	Px;Sx+x sek.	**P2; C1+5, C3+5, C6+5, C8+5** **P3; C1+5, C3+5, C6+5, C8+5** **P4; C1+5, C3+5, C6+5, C8+5**
23	Flytt1, Starter	Px;Sx+x sek.	
24	Flytt2, Starter	Px;Sx+x sek.	**P2; C1+10, C3+10, C6+10, C8+10** **P3; C1+10, C3+10, C6+10, C8+10** **P4; C1+10, C3+10, C6+10, C8+10**
25	Flytt2, Starter	Px;Sx+x sek.	

26	Status; måste grön	SGx	**SG1, SG3**
27	Status; kan grön	SGx	
28	Status; måste röd	SGx	
29	Status; kan röd	SGx	
30	Status; annan signal	Anl.x, SGx	
31	Vx1; SG måste grön	SGx	
32			
33	Vx1; SG till rött direkt	SGx-M/Fx sek.	
34	Vx1; SG till rött garanti	SGx-x sek.	
35	Vx1; SG till grönt	SGx-Mx sek.	
36	Vx2; SG måste grön	SGx	
37			
38	Vx2; SG till rött direkt	SGx-M/Fx sek.	
39	Vx2; SG till rött garanti	SGx-x sek.	
40	Vx2; SG till grönt	SGx-Mx sek.	
41	För; SG måste grön	SGx	
42			
43	För; SG förlängas	SGx-M/Fx sek.	**SG1 M=50, SG3 M=50**
44	För; SG till rött garanti	SGx-x sek.	
45			
46	Ny Time-Out SG grön	x sek.	
47	Sista-Steg **(Se PRP16)**	Px;Sx+x sek.	**P2; C2+18** **P3; C2+18** **P4; C2+18**
48	Till rött vid avslut	SGx-M/Fx sek.	
49	Till grönt vid avslut	SGx-M sek.	
50	Förhindra egen funktion	x sek.	

Anmärkningar:

Anl.Nr:	**01**	**Address:**	**Testvägen-Provgatan**		
Funktion nr:	**2**	**Projektör:**	**Olle Proj**		
Ant. funktioner:	**4**	**Beställare**	**Komunen**		
Tillfart:	**F3**	**Datum:**	**1996-06-27**	**Busslinje:**	**Alla**

1	Typ av funktion	BF,ÅTS.....	**BF**
2			
3	Aktiv i tidplan	Px	**P2, P3, P4**
4			
5			
6	Förlängande detektor	SGx Mx,Fx sek.	**SG1 M=0.1**
7	Räknare	Rx	**R1**
8	Anmälande detektor	Det x	**BD3**
9	Avanmälande detektor	Det x	**BD4**
10			
11	Time-Out	x sek.	**20 sek**
12	Tillfart	SGx	**F3**
13	Motsatt räknare	Rx	**R1**
14			
15			
16	Lucka	Px;Sx+x sek.	**P2; C1+40-C2+0** **P3; C1+40-C2+0** **P4; C1+40-C2+0**
	Lucka NyBuss	Px;Sx+x sek.	**P2; C1+40-C2+5** **P3; C1+40-C2+5** **P4; C1+40-C2+5**
	Lucka Motriktad	Px;Sx+x sek.	**P2; C1+40-C2+10** **P3; C1+40-C2+10** **P4; C1+40-C2+10**
17	Prio.beg. från start	Fx-x sek.	
18	Prio.beg. från start	Fx-x sek.	
19	Prio.beg. från avslut	Fx-x sek.	**F3-20, F4-20**
20	Prio.beg. från avslut	Fx-x sek.	
21	Lucka Flytt1	Px;Sx+x sek.	**P2; C1+40-C2+10** **P3; C1+40-C2+10** **P4; C1+40-C2+10**
	Lucka Flytt2	Px;Sx+x sek.	**P2; C2+10-C2+18** **P3; C2+10-C2+18** **P4; C2+10-C2+18**
22	Flytt1, Starter	Px;Sx+x sek.	**P2; C1+5, C3+5, C6+5, C8+5** **P3; C1+5, C3+5, C6+5, C8+5** **P4; C1+5, C3+5, C6+5, C8+5**
23	Flytt1, Starter	Px;Sx+x sek.	
24	Flytt2, Starter	Px;Sx+x sek.	**P2; C1+10, C3+10, C6+10, C8+10** **P3; C1+10, C3+10, C6+10, C8+10** **P4; C1+10, C3+10, C6+10, C8+10**
25	Flytt2, Starter	Px;Sx+x sek.	

26	Status; måste grön	SGx	**SG1, SG3**
27	Status; kan grön	SGx	
28	Status; måste röd	SGx	
29	Status; kan röd	SGx	
30	Status; annan signal	Anl.x, SGx	
31	Vx1; SG måste grön	SGx	
32			
33	Vx1; SG till rött direkt	SGx-M/Fx sek.	
34	Vx1; SG till rött garanti	SGx-x sek.	
35	Vx1; SG till grönt	SGx-Mx sek.	
36	Vx2; SG måste grön	SGx	
37			
38	Vx2; SG till rött direkt	SGx-M/Fx sek.	
39	Vx2; SG till rött garanti	SGx-x sek.	
40	Vx2; SG till grönt	SGx-Mx sek.	
41	För; SG måste grön	SGx	**SG1, SG3**
42			
43	För; SG förlängas	SGx-M/Fx sek.	**SG1 M=50, SG3 M=50**
44	För; SG till rött garanti	SGx-x sek.	
45			
46	Ny Time-Out SG grön	x sek.	
47	Sista-Steg **(Se PRP16)**	Px;Sx+x sek.	**P2; C2+18** **P3; C2+18** **P4; C2+18**
48	Till rött vid avslut	SGx-M/Fx sek.	
49	Till grönt vid avslut	SGx-M sek.	
50	Förhindra egen funktion	x sek.	

Anmärkningar:

Anl.Nr:	**01**	**Address:**	**Testvägen-Provgatan**		
Funktion nr:	**3**	**Projektör:**	**Olle Proj**		
Ant. funktioner:	**4**	**Beställare**	**Komunen**		
Tillfart:	**F1**	**Datum:**	**1996-06-27**	**Busslinje:**	**Alla**

1	Typ av funktion	BF,ÅTS.....	**ÅTS**
2			
3	Aktiv i tidplan	Px	**P2, P3, P4**
4			
5			
6	Förlängande detektor	SGx Mx,Fx sek.	**SG1 M=0.1**
7	Räknare	Rx	**R1**
8	Anmälande detektor	Det x	
9	Avanmälande detektor	Det x	
10			
11	Time-Out	x sek.	
12	Tillfart	SGx	
13	Motsatt räknare	Rx	**R2**
14			
15			
16	Lucka	Px;Sx+x sek.	**P2; C2+0-C2+5** **P3; C2+0-C2+5** **P4; C2+0-C2+5**
	Lucka NyBuss	Px;Sx+x sek.	**P2; C2+0-C2+10** **P3; C2+0-C2+10** **P4; C2+0-C2+10**
	Lucka Motriktad	Px;Sx+x sek.	
17	Prio.beg. från start	Fx-x sek.	
18	Prio.beg. från start	Fx-x sek.	
19	Prio.beg. från avslut	Fx-x sek.	
20	Prio.beg. från avslut	Fx-x sek.	
21	Lucka Flytt1	Px;Sx+x sek.	**P2; C1+40-C2+10** **P3; C1+40-C2+10** **P4; C1+40-C2+10**
	Lucka Flytt2	Px;Sx+x sek.	**P2; C2+10-C2+18** **P3; C2+10-C2+18** **P4; C2+10-C2+18**
22	Flytt1, Starter	Px;Sx+x sek.	**P2; C1+5, C3+5, C6+5, C8+5** **P3; C1+5, C3+5, C6+5, C8+5** **P4; C1+5, C3+5, C6+5, C8+5**
23	Flytt1, Starter	Px;Sx+x sek.	
24	Flytt2, Starter	Px;Sx+x sek.	**P2; C1+10, C3+10, C6+10, C8+10** **P3; C1+10, C3+10, C6+10, C8+10** **P4; C1+10, C3+10, C6+10, C8+10**
25	Flytt2, Starter	Px;Sx+x sek.	

26	Status; måste grön	SGx	
27	Status; kan grön	SGx	
28	Status; måste röd	SGx	**SG1, SG2, SG4, SG5, SG7**
29	Status; kan röd	SGx	
30	Status; annan signal	Anl.x, SGx	
31	Vx1; SG måste grön	SGx	
32			
33	Vx1; SG till rött direkt	SGx-M/Fx sek.	
34	Vx1; SG till rött garanti	SGx-x sek.	
35	Vx1; SG till grönt	SGx-Mx sek.	
36	Vx2; SG måste grön	SGx	
37			
38	Vx2; SG till rött direkt	SGx-M/Fx sek.	**SG2, SG4, SG5, SG7**
39	Vx2; SG till rött garanti	SGx-x sek.	
40	Vx2; SG till grönt	SGx-Mx sek.	**SG1**
41	För; SG måste grön	SGx	**SG1**
42			
43	För; SG förlängas	SGx-M/Fx sek.	**SG1 M=50**
44	För; SG till rött garanti	SGx-x sek.	
45			
46	Ny Time-Out SG grön	x sek.	
47	Sista-Steg **(Se PRP16)**	Px;Sx+x sek.	**P2; C2+18** **P3; C2+18** **P4; C2+18**
48	Till rött vid avslut	SGx-M/Fx sek.	**SG1**
49	Till grönt vid avslut	SGx-M sek.	**SG2, SG4, SG5, SG7**
50	Förhindra egen funktion	x sek.	

Anmärkningar:

Anl.Nr:	**01**	**Address:**	**Testvägen-Provgatan**		
Funktion nr:	**4**	**Projektör:**	**Olle Proj**		
Ant. funktioner:	**4**	**Beställare**	**Komunen**		
Tillfart:	**F1**	**Datum:**	**1996-06-27**	**Busslinje:**	**Alla**

1	Typ av funktion	BF,ÅTS.....	**ÅTS**
2			
3	Aktiv i tidplan	Px	**P2, P3, P4**
4			
5			
6	Förlängande detektor	SGx Mx,Fx sek.	**SG3 M=0.1**
7	Räknare	Rx	**R2**
8	Anmälande detektor	Det x	
9	Avanmälande detektor	Det x	
10			
11	Time-Out	x sek.	
12	Tillfart	SGx	
13	Motsatt räknare	Rx	
14			
15			
16	Lucka	Px;Sx+x sek.	**P2; C2+0-C2+5** **P3; C2+0-C2+5** **P4; C2+0-C2+5**
	Lucka NyBuss	Px;Sx+x sek.	**P2; C2+0-C2+5** **P3; C2+0-C2+5** **P4; C2+0-C2+5**
	Lucka Motriktad	Px;Sx+x sek.	
17	Prio.beg. från start	Fx-x sek.	
18	Prio.beg. från start	Fx-x sek.	
19	Prio.beg. från avslut	Fx-x sek.	
20	Prio.beg. från avslut	Fx-x sek.	
21	Lucka Flytt1	Px;Sx+x sek.	**P2; C1+40-C2+10** **P3; C1+40-C2+10** **P4; C1+40-C2+10**
	Lucka Flytt2	Px;Sx+x sek.	**P2; C2+10-C2+18** **P3; C2+10-C2+18** **P4; C2+10-C2+18**
22	Flytt1, Starter	Px;Sx+x sek.	**P2; C1+5, C3+5, C6+5, C8+5** **P3; C1+5, C3+5, C6+5, C8+5** **P4; C1+5, C3+5, C6+5, C8+5**
23	Flytt1, Starter	Px;Sx+x sek.	
24	Flytt2, Starter	Px;Sx+x sek.	**P2; C1+10, C3+10, C6+10, C8+10** **P3; C1+10, C3+10, C6+10, C8+10** **P4; C1+10, C3+10, C6+10, C8+10**
25	Flytt2, Starter	Px;Sx+x sek.	

26	Status; måste grön	SGx	
27	Status; kan grön	SGx	
28	Status; måste röd	SGx	**SG3, SG2, SG4, SG5, SG7**
29	Status; kan röd	SGx	
30	Status; annan signal	Anl.x, SGx	
31	Vx1; SG måste grön	SGx	
32			
33	Vx1; SG till rött direkt	SGx-M/Fx sek.	
34	Vx1; SG till rött garanti	SGx-x sek.	
35	Vx1; SG till grönt	SGx-Mx sek.	
36	Vx2; SG måste grön	SGx	
37			
38	Vx2; SG till rött direkt	SGx-M/Fx sek.	**SG2, SG4, SG5, SG7**
39	Vx2; SG till rött garanti	SGx-x sek.	
40	Vx2; SG till grönt	SGx-Mx sek.	**SG3**
41	För; SG måste grön	SGx	**SG3**
42			
43	För; SG förlängas	SGx-M/Fx sek.	**SG3 M=50**
44	För; SG till rött garanti	SGx-x sek.	
45			
46	Ny Time-Out SG grön	x sek.	
47	Sista-Steg **(Se PRP16)**	Px;Sx+x sek.	**P2; C2+18** **P3; C2+18** **P4; C2+18**
48	Till rött vid avslut	SGx-M/Fx sek.	**SG3**
49	Till grönt vid avslut	SGx-M sek.	**SG2, SG4, SG5, SG7**
50	Förhindra egen funktion	x sek.	

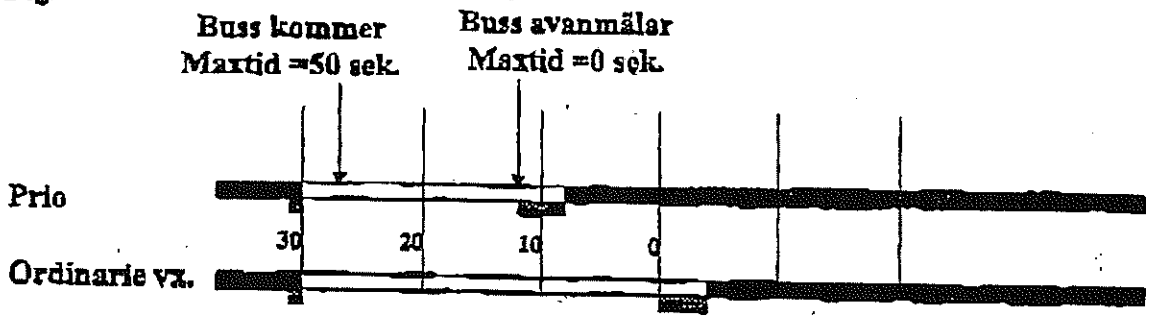
Anmärkningar:

3. Komplettering av PRIPTS-logiken för att underlätta prioritering i oberoende styrning.

3.1 Förlägning. (Funktion PG 25-GNo-MAX)

I dagens logik ges busstillfarten en ny maxtid som bussen kan förlänga. Denna tid ges till gruppen oberoende av hur mycket ordinarie maxtid som finns kvar. Konsekvensen blir att när bussen avanmäls nollställs maxtiden oberoende av ordinarie maxtid.

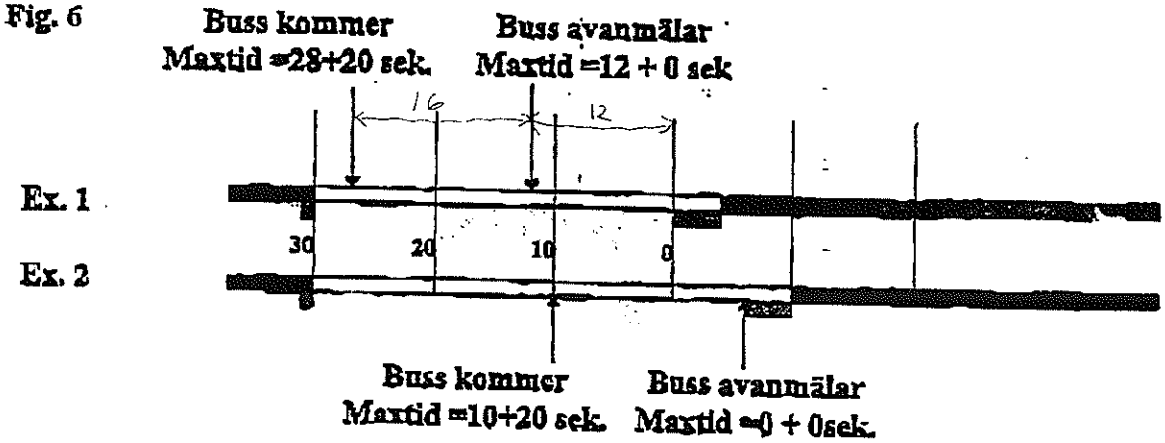
Fig. 5



Förslag till ändring:

En ny funktion 29-GNo-MaxAdd som adderar en valfri tid till kvarvarande ordinarie maxtid.

Fig. 6



Ex1.(se fig.6) När bussen anmäls laddas ordinarie kvarvarande maxtid in i en tidräknare T1. Den nya maxtiden blir summan av ordinarie maxtid T1 och den tid som vi har programmerad i PRP 43. När bussen kommer är kvarvarande ordinarie maxtid 28 sekunder och maxtidstillskottet 20 sekunder (PRP43). Maxtidsräknaren laddas med 28+20 sekunder. Vid avanmälan laddas maxtidsräknaren med resterande tid från T1 adderad med angiven tid från PRP48 (till rött vid EOF).

Programmeras på följande sätt:

Ex.1			
PRP43 V Ext (r imm)	29-GNo-20	Maxtid= T1+MaxAdd= 28+20	(Anmälan)
PRP48 red at EOF	29-GNo-0	Maxtid= T1+MaxAdd= 12+0	(Avanm.)
Ex.2			
PRP43 V Ext (r imm)	29-GNo-20	Maxtid= T1+MaxAdd= 10+20	(Anmälan)
PRP48 red at EOF	29-GNo-0	Maxtid= T1+MaxAdd= 0+0	(Avanm.)

3.2 Ny buss och ny buss motriktad. (Parameter PR 16)

Vid samordnad styrning kan man bestämma om ytterligare en ny buss eller en ny allierad buss skall tillåtas förlänga en pågående funktion. Detta går bra eftersom man har en fast referensram i omloppet och därför kan ange fönster för tillåtna nya bussar genom att definera "Lucka Ny" (New vehicle registering window definition, own direction) och Lucka Ny Motriktad (New vehicle registering window definition, opposite direction) så att sannolikheten är stor att dessa bussar skall hinna passera korsningen före funktionens sista tidsteg.

Vid oberoende styrning finns ingen sådan fast referensram. Detta innebär att det inte går att förhindra att ny buss som kommer sent får förlänga resterande tid fast den inte räcker till för att bussen skall hinna genom korsningen.

Förslag till ändring:

En ny funktion 30- **DriOwn-DriOpp** (körtid för ny buss och körtid för alliarad buss) som kontrollerar att resterande maxtid (T1+MaxAdd) för gruppen är större en tiden **DriOwn** eller **DriOpp** innan bussen godkännas.

3.3 Slå av lägestyrningen.

En ny funktion **31-SRing-SRing** som slår av lägesstyrningen (Disable stage ring) när bussen har godkänts (Anges i PRP3). Detta görs för att förhindra att konflikterande order ges till grupperna. **Lägesstyrningen** aktiveras när de grupper som är angivna i **PR49 grn at EOF** har blivit gröna (Enable stage ring).

3.4 Status läge.

En ny status **32-SS-00** som visar vilket läge som är aktivt (Läge SS måste vara aktivt för att funktionen skall vara sann). Anges i PRP 26-29. I dag kan man bara kontrollera om signalgrupper är röda eller gröna och det räcker inte till för att bestämma vilka grupper som är ifärd med att växla till grönt. Det är nödvändigt att veta vilket läga man befinner sig i för att t.ex. en ÅTS skall få träda i kraft.

3.5 Priority skall ha högre prioritet en PRIPTS.

Vid brandutryckning och tågprioritering skall funktionen BP33-34 PRIORITY A1 och A2 ha högre prioritet en PRIPTS-logiken. PRIORITY skall omedelbart avsluta PRIPTS-logiken.

3.6. Slå av och på "tidplansholdningen" efter prioritering.

Idag förhindras tidplansbyte i 2 minuter efter en prioritering. Det borde finnas en variabel som förhindrade tidplansholdningen. Funktionen **32-PLNo-PLNo** anges ~~i PRT~~ PRP3 och bestämmer att tidplansbyte mellan t.ex. TP9 och TP10 (32-9-10) skall ske utan fördröjning. Detta gäller oftast vid länkning.