




Traffic
System Development
Consultancy


Programmeer Voorschrift IT&T Kwaliteitscentrale

(Versie 3 - Vialis/TPA)

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	006		Programmeer Voorschrift IT&T Kwaliteitscentrale 3 - TPA	RKa

Inhoudsopgave


1 Document historie	3
2 Inleiding	4
3 Programmeer voorschrift	5
3.1 Korte variant.....	5
3.2 Lange variant.....	6
3.3 CCOL.....	7
3.4 RWS-C.....	7
4 Applicatie specifieke parameters	8
4.1 Definities.....	8
4.1.1 <i>Prioriteits ingreep</i>	8
4.1.2 <i>Bijzondere ingreep</i>	8
4.1.3 <i>Snelheids meting</i>	8
4.1.4 <i>Max groen bereikt</i>	8
4.1.5 <i>Module afloop start</i>	8
4.1.6 <i>Alternatieve groen realisatie</i>	8
4.2 Parameter opbouw.....	8
4.3 Selectieve berichten.....	13
4.3.1 <i>KAR</i>	13
4.3.2 <i>DS-module</i>	14
4.3.3 <i>selectieve detectie lussen</i>	14
5 Voorbeelden	15
5.1 CCOL.....	15
5.1.1 <i>Definitie bestand</i>	15
5.1.2 <i>Instellingen</i>	15
5.1.3 <i>Regel bestand</i>	16

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	006		Programmeer Voorschrift IT&T Kwaliteitscentrale 3 - TPA	RKa

1 Document historie

Noodzaak voor een derde versie van het programmeervoorschrift is een functionele wijziging van de LTC[®] / ViTrac van Vialis/TPA. De wijziging stelt de gebruiker onder andere in staat om meerdere MV bestanden in de regelaar aan te maken t.b.v. de IT&T Kwaliteitscentrale[®].


Wijziging door:	Datum:	Omschrijving:
RKa	16-10-2001	Initiële aanmaak
RKa	05-09-2003	Document apart genomen uit MV-module interface doc.
RKa	24-09-2003	Enkele onjuiste benamingen gewijzigd
RKa	08-03-2005	Aangepast van versie 1 naar versie 3
RKa	24-01-2006	Toevoegingen t.b.v. ondersteuning CVN-3
RKa	28-01-2009	Verduidelijking logging selectieve detectie en KAR

	Datum 28-01-09	Versie 006	Status	Documentnaam Programmeer Voorschrift IT&T Kwaliteitscentrale 3 - TPA	Auteur RKa
---	--------------------------	----------------------	---------------	---	----------------------

2 Inleiding

Dit document is van toepassing voor de aanmaak van MV-files waarbij gebruik wordt gemaakt van het door IT&T geleverde **CCOL- of RWS-C object** (implementatie van de MV-file boven de interface) **versie 3** (in gebruik bij Vialis/TPA). De taak van het object is het verzamelen van alle benodigde gegevens uit de CVN-interface om met deze informatie op de voorgeschreven wijze de MV-file aan te maken.

Deze nieuwe beschrijving is noodzakelijk ten behoeve van ondersteuning van CVN-C interface versie 3.

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	006		Programmeer Voorschrift IT&T Kwaliteitscentrale 3 - TPA	RKa

3 Programmeer voorschrift

Dit programmeer voorschrift beschrijft de wijze waarop de applicatie programmeur de applicatie afhankelijke informatie dient aan te leveren op de CVN-interface. Het voordeel van een dergelijke aanpak is dat vervolgens met een standaard module, applicatie afhankelijke gegevens vergaard kunnen worden, nodig voor generatie van berichten voor de kwaliteitscentrale. Alle beschreven parameters dienen te worden opgenomen, ook al maakt de regeling er geen gebruik van.

Er zijn twee varianten: een korte en een lange variant. De korte variant bevat de minimale parameter toevoegingen aan het regelprogramma om het logproces (MV-file) te kunnen beïnvloeden.

Alle wijzigingen in de PARM1, PARM2, IS en GUS/WUS buffer van de CVN-interface worden gelogd. Het is derhalve niet nodig om in dit programmeervoorschrift ruimte te definiëren voor vrij te definiëren parameters die van invloed kunnen zijn op resultaten van de PI-berekeningen. De betreffende parameters kunnen gewoon in de standaard parameters worden opgenomen en bij presentatie van gecalculerde waarden worden getoond.

3.1 Korte variant

In de korte variant wordt gebruik gemaakt van de minimaal benodigde parameters (zie onder hoofdstuk "parameter opbouw" (4.2) de punten 1 t/m 4).

Het voordeel is dat in de applicatie niet meer dan vier parameters, 1 include file en 1 functie call hoeft te worden toegevoegd.

Het nadeel is dat het object niet kijkt naar gegevens welke zijn vermeld onder de punten 5 t/m 9 van hoofdstuk 4.2.

Dat betekent dat, indien bijvoorbeeld een applicatie specifieke bijzondere ingreep wordt opgenomen, de lange variant noodzakelijk is.

Voor de korte variant dient het volgende in de applicatie te worden uitgevoerd:

CCOL voorbeeld:


1. Voeg 4 parameters toe in de definitie file achter de reeds bestaande parameters:

(index nummers zijn applicatie afhankelijk)

```
#define prm_kwc_tijd      45 /* zie punt 4 onder 'parameter opbouw' */
#define prm_kwc_sch      46 /* zie punt 3 onder 'parameter opbouw' */
#define prm_kwc_version  47 /* zie punt 2 onder 'parameter opbouw' */
#define prm_kwc_id       48 /* zie punt 1 onder 'parameter opbouw' */
#define PRMMAX           49
```

2. Geef de parameters een naam en een waarde in de control_parameters() functie:

```
PRM_code[prm_kwc_tijd]   = "KWC_TIJD";      PRM[prm_kwc_tijd]     = 60;
PRM_code[prm_kwc_sch]   = "KWC_SCH";        PRM[prm_kwc_sch]     = 7;
PRM_code[prm_kwc_version] = "KWC_VERSION";  PRM[prm_kwc_version] = 3;
PRM_code[prm_kwc_id]    = "KWC_ID";         PRM[prm_kwc_id]      = 32492;
/* 32492 is het nummer bij het verkorte programmeervoorschrift */
```

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	006		Programmeer Voorschrift IT&T Kwaliteitscentrale 3 - TPA	RKa

3. Include in de regelfile de header "pi_ccol.h" en doe een call naar PI_Save() in de functie 'system_application()'

RWS-C voorbeeld:

1. Voeg 4 parameters toe in de crapdef.h file achter de reeds bestaande extra geheel getal parameters:

(index nummers zijn applicatie afhankelijk)

```
#define KWC_TIJD          3 /* zie punt 4 onder 'opbouw parameters' */
#define KWC_SCH          4 /* zie punt 3 onder 'opbouw parameters' */
#define KWC_VERSION      5 /* zie punt 2 onder 'opbouw parameters' */
#define KWC_ID           6 /* zie punt 1 onder 'opbouw parameters' */
#define NUMEGGP          7
```

2. Geef de parameters een waarde in de EPARMS_tabel (bestand craptab.c):

```
KWC_TIJD,          6.0,
KWC_SCH,           0.7,
KWC_VERSION,      0.3,
KWC_ID,            3249.2,
STOP };
```

/* 32492 is het nummer bij het verkorte programmeervoorschrift */

3. Include in crapcod.c de header "pi_rwsc.h" en doe een call naar PI_Save() in de functie 'overige_voorwaarden()'.


3.2 Lange variant

Naast datgene wat is beschreven onder de "korte variant" (met als aanpassing dat het getal "32492" nu "32493" moet zijn) is er de gelegenheid meer gegevens te verzamelen en de MV-file van buiten af te beïnvloeden voor wat betreft het wel/niet loggen van data van bepaalde fasecycli.

De programmeur moet hierbij voor alle punten 1 t/m 9 onder het hoofdstuk "Opbouw van de parameters" parameters toevoegen, ook al maakt de regeling er geen gebruik van.

Dit programmeer voorschrift beschrijft de wijze waarop de applicatie programmeur de applicatie afhankelijke informatie dient aan te leveren op de CVN-interface. Het voordeel van een dergelijke aanpak is dat vervolgens met een standaard module applicatie afhankelijke gegevens vergaard kunnen worden, nodig voor generatie van berichten voor de kwaliteitscentrale. Alle beschreven parameters dienen te worden opgenomen, ook al maakt de regeling er geen gebruik van.

Alle wijzigingen in de PARM1, PARM2, IS en GUS/WUS buffer van de CVN-interface kunnen worden gelogd. Het is derhalve niet noodzakelijk om in dit programmeervoorschrift ruimte te definiëren voor vrij te definiëren parameters die van invloed kunnen zijn op resultaten van de MV-berekeningen. De betreffende parameters kunnen gewoon in de standaard parameters worden opgenomen en bij presentatie van gecalculerde waarden worden getoond.

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	006		Programmeer Voorschrift IT&T Kwaliteitscentrale 3 - TPA	RKa

3.3 CCOL

De kwaliteitscentrale houdt bij berekening van de verschillende gegevens rekening met de op dat moment ingestelde *maximum* groentijd. In een CCOL regeling wordt in het algemeen met verlenggroentijden gewerkt. Bij gebruik van maximumgroentijden in een CCOL regeling wordt derhalve de maximum groentijden meestal in timers of parameters geplaatst.


Voor een correcte werking van de kwaliteitscentrale is het noodzakelijk dat de verlenggroentijd voor een bepaalde fasecyclus overeenkomt met de op dat moment geldende maximumgroentijd vermindert met de vastgroentijd van de richting. Dit kan bijvoorbeeld worden bewerkstelligd door het gebruik van de functie *max_star_groentijden_va_arg* uit de standaard CCOL module *stdfunc.c* die inhoudelijk de verlenggroentijd instelling van de fasecyclus aanpast naar de op dat moment geldende maximumgroentijd die via parameters wordt meegegeven.

Om de MV-module te gebruiken dient in de CCOL functie *system_application()* een call naar de functie *PI_Save()* gezet te worden. In hetzelfde applicatiebestand dient de header "pi_ccol.h" geinclude te worden (indien "pi_ccol.c" in het project wordt meegecompileerd, of "pi_ccol.obj" wordt gelinkt) òf "pi_ccol.c" (indien "pi_ccol.c" of "pi_ccol.obj" niet in het project wordt meegenomen).

3.4 RWS-C

Om de MV-module te gebruiken dient in de RWS-C functie *overige_voorwaarden()* een call naar de functie *PI_Save()* gezet te worden. In hetzelfde applicatiebestand (*crapcod.c*) dient de header "pi_rwsc.h" geinclude te worden (indien "pi_rwsc.c" in het project wordt meegecompileerd, of "pi_rwsc.obj" wordt gelinkt) òf "pi_rwsc.c" (indien "pi_rwsc.c" niet in het project wordt meegenomen).

Om in een RWS-C regeling toch de beschikking te hebben over logische namen voor alle IO van de regeling dient er voor iedere applicatie een set data structuren met namen gevuld te worden. Zie hiervoor het document "*MV Module RWSC versie 3.0.pdf*".

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	006		Programmeer Voorschrift IT&T Kwaliteitscentrale 3 - TPA	RKa

4 Applicatie specifieke parameters

4.1 Definities

4.1.1 Prioriteits ingreep

Een prioriteitsingreep voor het openbaar vervoer vindt plaats op een bepaalde richting, waarbij dit een positief effect heeft op de groenrealisatie van de betreffende richting (eerder groen, groen verlengen). Een melding wordt gegeven op het moment dat de beïnvloeding start.

Lange variant noodzakelijk.

4.1.2 Bijzondere ingreep

Op het moment dat een bijzondere ingreep **eindigt** wordt hier de tijdsduur van- en het type ingreep gelogd. Het gaat om niet openbaar vervoersingrepen. Voorbeelden zijn AHOB, brug en file. Indien het specifiek een ingreep voor één richting betreft wordt ook het signaalgroepnummer meegegeven.

Lange variant noodzakelijk.

4.1.3 Snelheids meting

De snelheid zoals die door de bijbehorende detector is gemeten.

Lange variant noodzakelijk.

4.1.4 Max groen bereikt

Een regeling heeft maxgroen bereikt indien verlenggroen (CCOL) of tweede hiaat (RWS-C) wordt verlaten terwijl er nog geen hiaat gevallen is, en indien dit niet ten gevolge van een afkap functie gebeurt (afkappen, versneld naar meeverleng groen).

Korte variant meestal voldoende (zie 4.2 punt 5).

4.1.5 Module afloop start

Het moment dat de kruispuntscyclus opnieuw start.

Korte variant voldoende.


4.1.6 Alternatieve groen realisatie

De boodschap (vlag) dat een richting alleen alternatief gerealiseerd is geweest. (melding bij einde groen)

Korte variant voldoende.

4.2 Parameter opbouw

Alle applicatie afhankelijke informatie nodig voor de Kwaliteitscentrale wordt aangeboden via vrije parameters op de CVN-interface (PRM voor CCOL, EPARM voor RWS-C) *achter* de parameters nodig voor de applicatie, d.w.z. de parameters voor de Kwaliteitscentrale staan altijd achteraan. Indien een regeling voor uitbreiding over n vrije parameters beschikte en voor implementatie van de applicatie afhankelijke parameters voor de MV-module daarvoor k parameters moet toevoegen, zal de buffer $n+k$ vrije parameters bevatten (PRMMAX of

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	006		Programmeer Voorschrift IT&T Kwaliteitscentrale 3 - TPA	RKa

NUMEGGP = n+k). Het aantal parameters is afhankelijk van het aantal selectieve in- en uitgangen en van het aantal snelheids detectie lussen.

De naamgeving is vrij te bepalen. De module kijkt naar de positie van een parameter op de interface en is derhalve niet afhankelijk van vaste namen.

De volgende beschrijving geeft de locatie en betekenis van parameters aan **van achter (de laatste parameter op de interface) naar voren**.

Naamgeving voor parameters overgenomen uit voorbeeld (zie hoofdstuk 5.1).

1. De laatste parameter (PRM KWCID) in de interface geeft aan of er een aanvulling is geweest volgens het 'Programmeer Voorschrift'.

Er zijn twee varianten: een **korte** en een **lange** variant. Het verschil daarin wordt aangegeven door de waarde in deze parameter. Bij de lange variant dient de waarde 32493 in de parameter geplaatst te worden. Bij de korte variant de waarde 32492.

De lange variant houdt in dat voor alle punten uit dit programmeervoorschrift parameters zijn gedefinieerd. Zie voor de korte variant de betreffende paragraaf.

De PI-module zal bij een andere waarde dan 32492 of 32493 op deze locatie ervan uitgaan dat er geen informatie voor de Kwaliteitscentrale wordt aangeboden op de CVN-interface.

Resultaat:

CIF_PARM1[laatste_prm] = 32493 (lange variant) òf

CIF_PARM1[laatste_prm] = 32492 (korte variant)

2. De op een na laatste parameter (PRM KWCVERSIE) dient het versienummer van het gebruikte programmeer voorschrift te bevatten. Huidige versie is versie 3.

Resultaat:

CIF_PARM1[laatste_prm - 1] = 3

3. Er wordt 1 parameter gereserveerd (PRM KWCSCH) voor configuratie van het log proces.

Indien een betreffend bit 1 is, dient de corresponderende actie uitgevoerd te worden. Uitvoering dient te stoppen bij het resetten van het betreffende bit.

Bit0 (1) : *vervallen*

Bit1 (2) : logging interne fasecyclus toestanden


Logging van interne fase toestanden is nog te verfijnen middels doorgifte van actuele VAG toestand van de fase middels de, bij de uitbreiding t.b.v. de kwaliteitscentrale behorende, fasecyclus parameters. Zie ook punt 5.

Bit2 (4) : logging alle parameter wijzigingen

Bit3 (8) : circulair loggen (0) / lineair loggen (1)

Er wordt altijd lineair binnen één bestand gelogd. De instelling van dit bit is een indicatie richting procesbesturing wat te doen bij verzoek tot openen van een nieuwe MV file, indien de beschikbare ruimte onvoldoende is voor een nieuw bestand.

Lineair loggen houdt in, dat op dat moment het verzoek wordt afgewezen, en pas gehonoreerd kan worden indien er bestanden verwijderd zijn (b.v. middels FTP). Circulair loggen houdt in, dat het oudste bestand wordt verwijderd, om ruimte te creëren voor het nieuwe bestand.

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	006		Programmeer Voorschrift IT&T Kwaliteitscentrale 3 - TPA	RKa

Bit4 (16) : normale header (0) / verkorte header (1)

Bit6 (64) : loggen maxgroen wijzigingen (0) / geen maxgroenwijzigingen loggen (1)

Bit7 (128) : loggen teller wijzigingen (0) / geen tellerwijzigingen loggen (1)

Bit8 (256) : verzoek tot éénmalig genereren van normale header, indien generatie van korte header is ingesteld.

Resultaat:

CIF_PARM1[laatste_prm - 2] = 6

- Er wordt 1 parameter gereserveerd (PRM KWCTIJD) voor aanduiding van maximale tijdsduur per MV bestand. Indien hier een waarde 0 wordt opgegeven, zal er gelogd worden totdat de beschikbare ruimte vol is. Bij een waarde anders dan 0 zal de aangegeven tijd in minuten gelogd worden. Na het ingestelde aantal minuten, of indien maximale beschikbare ruimte gevuld is, wordt de MV file afgesloten en direct getracht een nieuw bestand te openen.

Resultaat:

CIF_PARM1[laatste_prm - 3] = max duur MV registratie

- De parameters hiervoor geven fasecyclus afhankelijke zaken aan (PRM KWCxx). Hiervoor dienen net zoveel parameters te worden gereserveerd als er fasecycli zijn (FCMAX, NUMSG). De bits in een specifieke parameter geeft voor de betreffende fasecyclus de volgende zaken weer:

Bit0 (0x0001) : de richting heeft maxgroen bereikt.

Maxgroen bereikt wordt door module herkend als: einde (2e) verlenggroen én geen afkapfunctie en er wordt verkeer gemeten (EVG[] && !(Z[] || FM[]) && MK[]). Indien er geen gebruikt wordt gemaakt van MK[] functionaliteit binnen CCOL (maar van b.v. YV[]) gaat deze automatische herkenning dus niet juist, en kan de programmeur zelf via dit bit aangeven of maxgroen bereikt is.

Bit1 (0x0002) : de richting is alternatief gerealiseerd.

De module zal dit zelf automatisch herkennen. Opzetten van dit bit is derhalve niet noodzakelijk.

Bit2 (0x0004) : de richting kent een prioriteits ingreep.

Bit opzetten bij start van prioriteits ingreep (bijvoorbeeld op start hulpelement).


Bit3 (0x0008) : start aanvraag op de richting

De module zal dit zelf automatisch herkennen. Opzetten van dit bit is derhalve niet noodzakelijk.

Bit4+5+6 (0x0070) : actuele VAG (volgens IVER VAG afhandeling)

Voor het definiëren van een aparte interne fasecyclus voor de diverse VAG gebieden (1 .. 4), kan de programmeur via deze 3 bits (indien de fasecyclus zich op dat moment in een specifiek VAG gebied bevindt) dit doorgeven naar de MV module. Indien de fase zich niet in een specifiek VAG gebied bevindt (vastgroen¹, wachtgroen, meeverlenggroen) dient hier 0 te worden ingevuld.

1 Strict gezien starten zowel VAG1 als VAG2 op startgroen. Echter voor de fasenlog met interne fasecyclus toestanden is het duidelijker om bij deze 3 overlappende gebieden een prioriteit te geven aan vastgroen, VAG1 op te geven, indien vastgroen is beëindigd en VAG1 nog niet, en VAG2 indien zowel vastgroen als VAG1 zijn beëindigd.

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	006		Programmeer Voorschrift IT&T Kwaliteitscentrale 3 - TPA	RKa

Bit7 (0x0080) : de richting heeft maxgroen **niet** bereikt.

Maxgroen niet bereikt wordt door module herkend als: einde (2e) verlenggroen én geen afkapfunctie én er wordt ook geen verkeer meer gemeten (EVG[] && !(Z[] || FM[]) && ! MK[]). Indien er geen gebruikt wordt gemaakt van MK[] functionaliteit binnen CCOL (maar van b.v. YV[]) gaat deze automatische herkenning dus niet juist, en kan de programmeur zelf via dit bit aangeven of maxgroen niet bereikt is.

Bit8 (0x0100) : einde aanvraag op de richting (ingetrokken)

De module zal dit zelf automatisch herkennen. Opzetten van dit bit is derhalve niet noodzakelijk.

Bit9 (0x0200) : openbaar vervoer inmelding op fasecyclus

Bit10 (0x0400) : openbaar vervoer uitmelding op fasecyclus

...

Bit15 (0x8000) : schakelaar bit: GEEN MV gegevens genereren voor de betreffende fasecyclus

Indien een richting een bepaalde status heeft bereikt, dient de applicatie het betreffende bit in de parameter *1 applicatie ronde* de waarde 1 te geven, en dient vervolgens de bit-waarde weer terug te zetten op 0. Dit geldt niet voor de bits 4+5+6.

Resultaat (bij FC_{MAX} = n, fasecycli fc₀ .. fc_{n-1}):

CIF_PARM1[laatste_prm - 3 - FC_{MAX}] = gegevens eerste fasecyclus (fc₀)

CIF_PARM1[laatste_prm - 3 - FC_{MAX} + 1] = gegevens tweede fasecyclus (fc₁)

...

CIF_PARM1[laatste_prm - 3 - FC_{MAX} + (n-1)] = gegevens n-de fasecyclus (fc_{n-1})²

6. Voor bijzondere ingrepen worden 3 parameters gereserveerd.

De parameters dienen alle in 1 keer gevuld te worden bij het *einde* van een bijzondere ingreep (1 applicatie ronde).

De eerste parameter (PRM KWCBITYPE) dient voor het vastleggen van het type ingreep.

1 : AHOB

2 : Brug

3 : File

4 : HV

5 : Fixatie


6 : Opticom

> 6 : Overig (applicatie afhankelijke betekenis)

De tweede parameter (PRM KWCBITIID) geeft de tijdsduur aan dat de betreffende ingreep geduurd heeft in seconden.

De derde parameter (PRM KWCBIFC) dient voor het vastleggen van de fasecyclus waarop de

² [laatste_prm - 3 - FC_{MAX} + (n - 1)] waarbij FC_{MAX} == n, is hetzelfde als [laatste_prm - 4]

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	006		Programmeer Voorschrift IT&T Kwaliteitscentrale 3 - TPA	RKa

ingreep plaats heeft gevonden. Indien er geen richting aan de ingreep verbonden is, dient men hier 255 (0xFF) in te vullen.

Resultaat:

CIF_PARM1[laatste_prm - 3 - FCMAX - 3] = type ingreep

CIF_PARM1[laatste_prm - 3 - FCMAX - 2] = ingreep tijdsduur

CIF_PARM1[laatste_prm - 3 - FCMAX - 1] = ingreep fasecyclus

7. Er worden, een applicatie afhankelijk aantal, parameters gereserveerd voor selectieve in en uitmeldingen.

De eerste parameter voor de parameters uit punt 6 (PRM KWCNRVCUIT) geeft aan hoeveel selectieve uitmeldings signalen er in de applicatie gebruikt worden. De parameter daarvoor (PRM KWCNRVCIN) geeft het aantal selectieve inmeldings signalen.

Hiervóór worden er per selectief ingangssignaal (in- of uitmelding) een parameters gereserveerd t.b.v. opgeven locatie in IS-buffer of DS-buffer waar de betreffende ingang zich bevindt.

De module controleert aan de hand van de waarde van DS_MAX of betreffende indici bij de IS-buffer danwel bij de DS-buffer horen. *Er is geen mix toegestaan.*

De parameters zijn als volgt gegroepeerd: index-parameters selectieve inmelding ingang, index-parameters selectieve uitmelding ingang, parameter aantal selectieve inmelders, parameter aantal selectieve uitmelders.

Resultaat bij p selectieve inmelders en q selectieve uitmelders :

CIF_PARM1[laatste_prm - 8 - FCMAX - $q - p$] = index 1^e sel. inmelder

...

CIF_PARM1[laatste_prm - 8 - FCMAX - $q - 1$] = index p^e sel. inmelder

CIF_PARM1[laatste_prm - 8 - FCMAX - q] = index 1^e sel. uitmelder

...

CIF_PARM1[laatste_prm - 8 - FCMAX - 1] = index q^e sel. uitmelder

CIF_PARM1[laatste_prm - 6 - FCMAX - 2] = aantal selectieve inmelder

CIF_PARM1[laatste_prm - 6 - FCMAX - 1] = aantal selectieve uitmelder

8. Er worden, een applicatie afhankelijk aantal, parameters gereserveerd voor snelheids meldingen.


De eerste parameter voor de parameters uit punt 7 (PRM KWCNRSNEL) geeft aan hoeveel snelheids ingangen er in de applicatie gebruikt worden. Voor ieder snelheids ingang wordt een parameters gedefinieerd t.b.v. opgeven locatie in IS buffer waar de betreffende snelheids melding ingang zich bevindt. *De codering die op de betreffende ingangen verwacht wordt, is zoals beschreven in de CVN-C interface versie 3, paragraaf 5.2.3.*

Resultaat bij p selectieve ingangen en q selectieve uitgangen en r snelheids ingangen:

CIF_PARM1[laatste_prm - 9 - FCMAX - $q - p - r$] = index 1^e snelheids ingang

...

CIF_PARM1[laatste_prm - 9 - FCMAX - $q - p - 1$] = index r^e snelheids ingang

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	006		Programmeer Voorschrift IT&T Kwaliteitscentrale 3 - TPA	RKa

CIF_PARM1[laatste_prm - 8 - FCMAX - q - p - 1] = aantal snelheids ingangen

9. Er worden, een applicatie afhankelijk aantal, parameters gereserveerd voor voertuig lengtedetectie meldingen.

De eerste parameter voor de parameters uit punt 8 (PRM KWCNRLLENGTH) geeft aan hoeveel lengte detectie meldingen er in de applicatie gebruikt worden. Voor ieder lengte detectie melding worden een parameters gedefinieerd t.b.v. opgeven locatie in IS buffer waar de betreffende lengte melding ingang zich bevindt. De verwachte eenheid van de meldingen is in decimeters.

Resultaat bij p selectieve ingangen, q selectieve uitgangen, r snelheids ingangen en s lengte meldings ingangen:

CIF_PARM1[laatste_prm - 10 - FCMAX - q - p - r - s] = index 1^e lengtemelding ingang

...

CIF_PARM1[laatste_prm - 10 - FCMAX - q - p - r - 1] = index s^e lengtemelding ingang

CIF_PARM1[laatste_prm - 9 - FCMAX - q - p - r - 1] = aantal snelheids ingangen

10. Er worden, een applicatie afhankelijk aantal, parameters gereserveerd voor voertuig classificatie meldingen.

De eerste parameter voor de parameters uit punt 9 (PRM KWCNRCLASS) geeft aan hoeveel classificatie ingangen er in de applicatie gebruikt worden. Voor ieder classificatie ingang wordt een parameters gedefinieerd t.b.v. opgeven locatie in IS buffer waar de betreffende ingang zich bevindt. Deze ingangen zijn alleen noodzakelijk indien de classificatie is uitgebreid ten opzichte van de CVN-C versie 3 codering (die worden uitgelezen bij snelheids ingangen), en in deze uitgebreide vorm kunnen worden aangeboden door de fabrikant.

Resultaat bij p selectieve inmelders, q selectieve uitmelders, r snelheids ingangen, s lengte detectie ingangen en t classificatie ingangen:

CIF_PARM1[laatste_prm - 11-FCMAX-q-p-r-s - t] = index 1^e classificatie ingang

...

CIF_PARM1[laatste_prm - 11-FCMAX-q-p-r-s - 1] = index t^e classificatie ingang

CIF_PARM1[laatste_prm - 11-FCMAX-q-p-r-s - 1] = aantal classificatie ingangen

4.3 Selectieve berichten


4.3.1 KAR

Kar berichten kunnen op 2 manieren aan de applicatie aangeboden worden:

1. Via CVN-interface CIF_DSI posities (CVN 3).
2. Via overige ingangen. Hierbij worden dan de laatste 37 overige ingangen op de interface gebruikt.

*Indien de berichten via overige ingangen worden doorgegeven aan de KAR-module, kan dit middels het toekennen binnen de applicatie van de waarde 1 aan de in de MV-module opgenomen variabele **has_kar_ing** worden aangegeven. Als gevolg hiervan zullen deze ingangen buiten beschouwing worden gelaten bij overige ingang gerelateerde event generatie.*

Indien de gegevens via de CVN_DSI buffer aan de applicatie worden doorgegeven is het niet

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	006		Programmeer Voorschrift IT&T Kwaliteitscentrale 3 - TPA	RKa

noodzakelijk elementen in de applicatie op te nemen om deze gegevens in de MV file te loggen. Alle informatie die aan de CVN_DSI buffer wordt aangeboden wordt altijd volledig in de MV-file opgenomen.

4.3.2 DS-module

Bij het gebruik van de CCOL DS-module kan een bericht van selectieve detectie op 2 manieren door de procesbesturing aan de DS-module worden doorgegeven. Dit is middels een bericht via een ringbuffer, die de DS-module vervolgens decodeert, danwel middels overige ingangen (8). In beide gevallen wordt het gedecodeerde bericht in DS_ variabelen geplaatst die de MV-module gebruikt voor event generatie.


De in de applicatie opgenomen DS-lusnummers worden in de MV-file bij overige ingangen geplaatst. De *sources* van de events behorende bij lusnummers uit de selectieve detectie buffer zullen dus starten bij ISMAX.

*Indien de berichten via overige ingangen worden doorgegeven aan de DS-module, kan dit middels het toekennen binnen de applicatie van de waarde 1 aan de in de MV-module opgenomen variabele **has_ds_ing** worden aangegeven. Als gevolg hiervan zullen deze ingangen buiten beschouwing worden gelaten bij overige ingang gerelateerde event generatie.*

4.3.3 selectieve detectie lussen

Een selectief detectiesysteem kan naast het gebruik van de DS module, de informatie van het systeem op twee manieren doorgeven aan de applicatie:

- In- en uitmeldingen via detectie ingangen. In de MV file komt hierdoor automatisch een detectie-op event bij in- of uitmelding openbaar vervoer. In de kwaliteitscentrale is dan aan het betreffende element de functionaliteit in- of uitmelding o.v. gekoppeld. Hiervoor hoeft niets in het programmeervoorschrift opgenomen te worden.
- In- en uitmeldingen worden doorgegeven middels CVN_DSI buffer. In het MV object wordt alles wat aan de CVN_DSI buffer wordt aangeboden in volledigheid gelogd in de MV file. Hiervoor hoeft niets in het programmeervoorschrift opgenomen te worden.

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	006		Programmeer Voorschrift IT&T Kwaliteitscentrale 3 - TPA	RKa

5 Voorbeelden

5.1 CCOL

5.1.1 Definitie bestand

```
#define kwcnrclass 39 /* aantal classificatie detectie ingangen */
#define kwcnrlength 40 /* aantal lengte detectie ingangen */
#define kwcnrnsnel 41 /* aantal snelheidsdetectie ingangen */
#define kwc02i 42 /* selectieve detectie ingang index ds02i */
#define kwc48i 43 /* selectieve detectie ingang index ds48i */
#define kwc49i 44 /* selectieve detectie ingang index ds49i */
#define kwc02u 45 /* selectieve detectie uitgang index ds02u */
#define kwc48u 46 /* selectieve detectie uitgang index ds48u */
#define kwc49u 47 /* selectieve detectie uitgang index ds49u */
#define kwcnrvcin 48 /* aantal selectieve inmelders */
#define kwcnrvcuit 49 /* aantal selectieve uitmelders */
#define kwcbitype 50 /* bijzondere ingreep type */
#define kwcbitijd 51 /* bijzondere ingreep tijdsduur */
#define kwcbifc 52 /* bijzondere ingreep fasecyclus */
#define kwc01 53 /* fasecyclus afhankelijke gegevens fc01 */
#define kwc02 54 /* fasecyclus afhankelijke gegevens fc02 */
#define kwc03 55 /* fasecyclus afhankelijke gegevens fc03 */
#define kwc05 56 /* fasecyclus afhankelijke gegevens fc05 */
#define kwc08 57 /* fasecyclus afhankelijke gegevens fc08 */
#define kwc09 58 /* fasecyclus afhankelijke gegevens fc09 */
#define kwc10 59 /* fasecyclus afhankelijke gegevens fc10 */
#define kwc11 60 /* fasecyclus afhankelijke gegevens fc11 */
#define kwc22 61 /* fasecyclus afhankelijke gegevens fc22 */
#define kwc26 62 /* fasecyclus afhankelijke gegevens fc26 */
#define kwc28 63 /* fasecyclus afhankelijke gegevens fc28 */
#define kwc48 64 /* fasecyclus afhankelijke gegevens fc48 */
#define kwc49 65 /* fasecyclus afhankelijke gegevens fc49 */
#define kwctijd 66 /* tijdsduur MV registratie per file */
#define kwcsch 67 /* programmeervoorschrift schakelaar */
#define kwcversie 68 /* programmeervoorschrift versie */
#define kwcid 69 /* programmeervoorschrift identifier */

#define PRMMAX 70 /* aantal parameters */
```

5.1.2 Instellingen

```
PRM(kwcnrclass , "KWCNRCLASS" , 0 , RO_type ); /* aantal classificatie detectie ingangen */
PRM(kwcnrlength , "KWCNRLENGTH" , 0 , RO_type ); /* aantal lengte detectie ingangen */
PRM(kwcnrnsnel , "KWCNRSNEL" , 0 , RO_type ); /* aantal snelheidsdetectie ingangen */
PRM(kwc02i , "KWC02I" , ds02i , RO_type ); /* selectieve detectie ingang index ds02i */
PRM(kwc48i , "KWC48I" , ds48i , RO_type ); /* selectieve detectie ingang index ds48i */
PRM(kwc49i , "KWC49I" , ds49i , RO_type ); /* selectieve detectie ingang index ds49i */
```

Datum	Versie	Status	Documentnaam	Auteur
28-01-09	006		Programmeer Voorschrift IT&T Kwaliteitscentrale 3 - TPA	RKa

```

PRM(kwc02u , "KWC02U" , ds02u , RO_type ); /* selectieve detectie uitgang index ds02u */
PRM(kwc48u , "KWC48U" , ds48u , RO_type ); /* selectieve detectie uitgang index ds48u */
PRM(kwc49u , "KWC49U" , ds49u , RO_type ); /* selectieve detectie uitgang index ds49u */
PRM(kwcnrvcin , "KWCNRVCIN" , 3 , RO_type ); /* aantal selectieve ingangen */
PRM(kwcnrvcuit , "KWCNRVCUIT" , 3 , RO_type ); /* aantal selectieve uitgangen */
PRM(kwcbitype , "KWCBITYPE" , 0 , 0 ); /* bijzondere ingreep type */
PRM(kwcbitijd , "KWCBITIJD" , 0 , 0 ); /* bijzondere ingreep tijdsduur */
PRM(kwcbifc , "KWCBIFC" , 255 , 0 ); /* bijzondere ingreep fasecyclus */
PRM(kwc01 , "KWC01" , 0 , 0 ); /* fasecyclus afhankelijke gegevens fc01 */
PRM(kwc02 , "KWC02" , 0 , 0 ); /* fasecyclus afhankelijke gegevens fc02 */
PRM(kwc03 , "KWC03" , 0 , 0 ); /* fasecyclus afhankelijke gegevens fc03 */
PRM(kwc05 , "KWC05" , 0 , 0 ); /* fasecyclus afhankelijke gegevens fc05 */
PRM(kwc08 , "KWC08" , 0 , 0 ); /* fasecyclus afhankelijke gegevens fc08 */
PRM(kwc09 , "KWC09" , 0 , 0 ); /* fasecyclus afhankelijke gegevens fc09 */
PRM(kwc10 , "KWC10" , 0 , 0 ); /* fasecyclus afhankelijke gegevens fc10 */
PRM(kwc11 , "KWC11" , 0 , 0 ); /* fasecyclus afhankelijke gegevens fc11 */
PRM(kwc22 , "KWC22" , 0 , 0 ); /* fasecyclus afhankelijke gegevens fc22 */
PRM(kwc26 , "KWC26" , 0 , 0 ); /* fasecyclus afhankelijke gegevens fc26 */
PRM(kwc28 , "KWC28" , 0 , 0 ); /* fasecyclus afhankelijke gegevens fc28 */
PRM(kwc48 , "KWC48" , 0 , 0 ); /* fasecyclus afhankelijke gegevens fc48 */
PRM(kwc49 , "KWC49" , 0 , 0 ); /* fasecyclus afhankelijke gegevens fc49 */
PRM(kwctijd , "KWCTIJD" , 60 , 0 ); /* tijdsduur MV registratie per file */
PRM(kwcsch , "KWCSCH" , 6 , 0 ); /* programmeervoorschrift schakelaar */
PRM(kwcversie , "KWCVERSIE" , 3 , RO_type ); /* programmeervoorschrift versie */
PRM(kwcid , "KWCID" , 32493 , RO_type ); /* programmeervoorschrift identifier */

```

5.1.3 Regel bestand

```

#ifdef AUTOMAAT
    #include "pi_ccol.h"
#else
    #include "pi_ccol.c"
#endif

void KwcApplication (void)
{
    count i;
    static mulv klok_fix;

    /* fasecyclus parameters */
    /* ----- */
    for (i = kwc01; i <= kwc49; ++i){
        PRM[i] &= 0x8000; /* reset alle bits (uitgezonderd schakelaar bit) */
    }

    /* richtingen met prioriteits realisatie */
    if (SH[hov02]) PRM[kwc02] |= 0x04; /* vecom prioriteits realisatie */

```

```
if (SH[hov48]) PRM[kwc48] != 0x04; /* vecom prioriteits realisatie */
if (SH[hov49]) PRM[kwc49] != 0x04; /* vecom prioriteits realisatie */

/* bijzondere ingrepen */
/* ----- */
PRM[kwcbitijd] = 0;
PRM[kwcbitype] = 0;
PRM[kwcbifc] = 255; /* geen koppeling met een specifiek fasecyclus */

if (CIF_IS[fix] && (klok_fix == 0))
    klok_fix = CIF_KLOK[CIF_SEC_TELLER];
else if ((klok_fix != 0) && !CIF_IS[fix])
{
    PRM[kwcbitype] = 5; /* overig */
    PRM[kwcbitijd] = CIF_KLOK[CIF_SEC_TELLER] - klok_fix;
    klok_fix = 0;
}
}

void system_application(void)
{
    ...
    KwcApplication();
    PI_save();
}
```