




Traffic
System Development
Consultancy


Programmeer Voorschrift IT&T Kwaliteitscentrale

(Versie 3)

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	005		Programmeer Voorschrift IT&T Kwaliteitscentrale 3	RKa

Inhoudsopgave

1 Document historie.....	3
2 Inleiding.....	4
3 Programmeer voorschrift.....	5
3.1 CCOL.....	5
3.2 RWS-C.....	5
4 Applicatie specifieke parameters.....	6
4.1 Definities.....	6
4.1.1 <i>Prioriteits ingreep</i>	6
4.1.2 <i>Bijzondere ingreep</i>	6
4.1.3 <i>Snelheids meting</i>	6
4.1.4 <i>Max groen bereikt</i>	6
4.1.5 <i>Module afloop start</i>	6
4.1.6 <i>Alternatieve groen realisatie</i>	6
4.2 Parameter opbouw.....	6
4.3 Selectieve berichten.....	10
4.3.1 <i>KAR</i>	10
4.3.2 <i>DS-module</i>	11
4.3.3 <i>selectieve detectie lussen</i>	11
5 MV-module interface functies t.b.v. applicaties.....	12
6 Voorbeelden.....	13
6.1 CCOL.....	13
6.1.1 <i>Definitie bestand</i>	13
6.1.2 <i>Instellingen</i>	13
6.1.3 <i>Regel bestand</i>	14


	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	005		Programmeer Voorschrift IT&T Kwaliteitscentrale 3	RKa

1 Document historie

Noodzaak voor een derde versie van het programmeervoorschrift is een functionele wijziging van de MV modules t.b.v. ondersteuning real-time toepassingen.

Daarnaast worden enkele nieuwe ID's door de nieuwe modules ondersteund, die middels sturing vanuit de applicatie aangestuurd dienen te worden.


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	24-11-2005	Aangepast van versie 2 naar versie 3
RKa	28-01-2009	Verduidelijking logging selectieve detectie en KAR

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	005		Programmeer Voorschrift IT&T Kwaliteitscentrale 3	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 CVN-interface) **versie 3** (in gebruik bij KoHartog, Siemens en Peek Traffic) 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, in combinatie met gewenste nieuwe events.

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	005		Programmeer Voorschrift IT&T Kwaliteitscentrale 3	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.

Indien de regelapplicatie geen bijzondere ingrepen, busingrepen, KAR, voertuig klassificatie, snelheids detectie en dergelijke kent, is het niet noodzakelijk dat de in hoofdstuk 4 genoemde parameters worden opgenomen.

Alle wijzigingen in de PARM1, PARM2, IS en GUS/WUS buffer van de CVN-interface kunnen 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 gecalculeerde waarden worden getoond.

3.1 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. Daarnaast dient voor de aanroep van *MvSave()* de waarden van het meetkriterium (*MK[]*) geldig te zijn.

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

3.2 RWS-C

Om de MV-module te gebruiken dient in de RWS-C functie *overige_voorwaarden()* een call naar de functie *MvSave()* gezet te worden. In hetzelfde applicatiebestand "mv_rwsc.c" geinclude te worden.

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	005		Programmeer Voorschrift IT&T Kwaliteitscentrale 3	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.

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.

4.1.3 Snelheids meting

De snelheid zoals die door de bijbehorende detector is gemeten.

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). Dit item *kan* door de MV-module automatisch gegenereerd worden, zonder applicatie toevoegingen.

4.1.5 Module afloop start

Het moment dat de kruispuntscyclus opnieuw start. Dit item kan door de MV-module bij een standaard CCOL/RWSC blokken afhandeling automatisch gegenereerd worden, zonder applicatie toevoegingen.


4.1.6 Alternatieve groen realisatie

De boodschap (vlag) dat een richting alleen alternatief gerealiseerd is geweest. (melding bij einde groen). Dit item kan door de MV-module automatisch gegenereerd worden, zonder applicatie toevoegingen.

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 NUMEP = $n+k$). Het aantal parameters is onder meer afhankelijk van het aantal selectieve in- en uitmeldings ingangen en van het aantal snelheids ingangen.

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.

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	005		Programmeer Voorschrift IT&T Kwaliteitscentrale 3	RKa

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

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

De mv-module zal bij een andere waarde dan 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

2. De op één na laatste parameter dient het versienummer van het gebruikte programmeer voorschrift te bevatten. Huidige versie is versie 3.

Resultaat:

CIF_PARM1[*laatste_prm-1*] = 3

3. De parameters hiervoor geven fasecyclus afhankelijke zaken aan. 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 kan 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	005		Programmeer Voorschrift IT&T Kwaliteitscentrale 3	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 gebruik 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 FCMAX = n, fasecycli $fc_0 .. fc_{n-1}$):

CIF_PARM1[laatste_prm-1-FCMAX] = gegevens eerste fasecyclus (fc_0)

CIF_PARM1[laatste_prm-1-FCMAX+1] = gegevens tweede fasecyclus (fc_1)

...

CIF_PARM1[laatste_prm-1-FCMAX+(n-1)] = gegevens n-de fasecyclus (fc_{n-1})

4. 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 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 geeft de tijdsduur aan dat de betreffende ingreep geduurd heeft in seconden.

De derde parameter dient voor het vastleggen van de fasecyclus waarop de ingreep plaats heeft gevonden. Indien er geen richting aan de ingreep verbonden is, dient men hier 255

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	005		Programmeer Voorschrift IT&T Kwaliteitscentrale 3	RKa

(0xFF) in te vullen.

Resultaat:

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

CIF_PARM1[laatste_prm-1-FCMAX-2] = ingreep tijdsduur

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

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

De eerste parameter voor de parameters uit punt 4 geeft aan hoeveel ingangssignalen t.b.v. selectieve uitmelding er in de applicatie gebruikt worden. De parameter daarvoor geeft het aantal ingangssignalen t.b.v. selectieve inmelding.

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 behoren bij de IS-buffer danwel DS-buffer horen. *Er is geen mix toegestaan.*

MV-events met wagennummer en/of lijnnummer informatie kan alleen worden gegenereerd indien de MV-module de selectieve meldingen via de DS-buffer (CCOL DS-module module of KAR-module) aangeboden krijgt.

Resultaat bij p selectieve inmelders en q selectieve uitmelders :

CIF_PARM1[laatste_prm-6-FCMAX-q-p] = index 1^e sel. inmeld.

...

CIF_PARM1[laatste_prm-6-FCMAX-q-1] = index p^e sel. inmeld.

CIF_PARM1[laatste_prm-6-FCMAX-q] = index 1^e sel. uitmeld.

...

CIF_PARM1[laatste_prm-6-FCMAX-1] = index q^e sel. uitmeld.

CIF_PARM1[laatste_prm-4-FCMAX-2] = aantal selectieve ingangen

CIF_PARM1[laatste_prm-4-FCMAX-1] = aantal selectieve uitgangen


6. Er worden, een applicatie afhankelijk aantal, parameters gereserveerd voor snelheids ingangen.

De eerste parameter voor de parameters uit punt 5 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-7-FCMAX-q-p-r] = index 1^e snelheids ingang

...

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	005		Programmeer Voorschrift IT&T Kwaliteitscentrale 3	RKa

CIF_PARM1[laatste_prm-7-FCMAX-q-p-1] = index r^e snelheids ingang

CIF_PARM1[laatste_prm-6-FCMAX-q-p-1] = aantal snelheids ingangen

7. Er worden, een applicatie afhankelijk aantal, parameters gereserveerd voor lengtedetectie ingangen.

De eerste parameter voor de parameters uit punt 6 geeft aan hoeveel lengte detectie ingangen er in de applicatie gebruikt worden. Voor ieder lengte detectie ingang wordt een parameters gedefinieerd t.b.v. opgeven locatie in IS buffer waar de betreffende ingang zich bevindt.

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

CIF_PARM1[laatste_prm-8-FCMAX-q-p-r-s] = index 1^e lengte detectie ingang

...

CIF_PARM1[laatste_prm-8-FCMAX-q-p-r-1] = index s^e lengte detectie ingang

CIF_PARM1[laatste_prm-7-FCMAX-q-p-r-1] = aantal lengte detectie ingangen

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

De eerste parameter voor de parameters uit punt 7 geeft aan hoeveel klassificatie ingangen er in de applicatie gebruikt worden. Voor ieder klassificatie 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 klassificatie uitgebreid is 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 klassificatie ingangen:

CIF_PARM1[laatste_prm-9-FCMAX-q-p-r-s-t] = index 1^e klassificatie ingang

...

CIF_PARM1[laatste_prm-9-FCMAX-q-p-r-s-1] = index t^e klassificatie ingang

CIF_PARM1[laatste_prm-8-FCMAX-q-p-r-s-1] = aantal klassificatie ingangen


4.3 Selectieve berichten

4.3.1 KAR

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

1. Via CVN-interface DS posities (CVN_DSI).
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

	Datum	Versie	Status	Documentnaam	Auteur
	28-01-09	005		Programmeer Voorschrift IT&T Kwaliteitscentrale 3	RKa

*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 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	005		Programmeer Voorschrift IT&T Kwaliteitscentrale 3	RKa

5 MV-module interface functies t.b.v. applicaties

Bij uitbreidingen van de MV-file met nieuwe ID's die (nog) niet door de module worden ondersteund, kan in zulke gevallen gebruik worden gemaakt van de volgende functies die in de MV-module zijn geïmplementeerd:

- *void MvWrite0(char id, char source);*
T.b.v. het schrijven van een ID zonder data.
- *void MvWrite1(char id, char source, char value);*
T.b.v. het schrijven van een ID met 1 byte ID afhankelijke data.
- *void MvWrite2(char id, char source, short value);*
T.b.v. het schrijven van een ID met 2 bytes ID afhankelijke data.
- *void MvWrite4(char id, char source, long value);*
T.b.v. het schrijven van een ID met 4 bytes ID afhankelijke data.

LET OP: Men kan niet zomaar een ID zelf verzinnen en deze met de gegeven functies aan een MV-file toevoegen. De ID's dienen door de kwaliteitscentrale herkenbaar te zijn, anders zal de resulterende MV-file niet leesbaar zijn!

Voor informatie of voor een bepaald gegeven een ID beschikbaar is, die niet gedekt wordt door dit programmeervoorschrift, maar wel eventueel met bovengenoemde functies af te handelen zou zijn, kan men contact opnemen met IT&T.

Datum	Versie	Status	Documentnaam	Auteur
28-01-09	005		Programmeer Voorschrift IT&T Kwaliteitscentrale 3	RKa

6 Voorbeelden

6.1 CCOL

6.1.1 Definitie bestand

```
#define kwcnrklas      52 /* aantal klassifikatie ingangen          */
#define kwcnrlength   53 /* aantal lengtedetectie ingangen          */
#define kwcnrnsnel    54 /* aantal snelheidsdetectie ingangen      */
#define kwc02i        55 /* selectieve detectie ingang index ds02i */
#define kwc48i        56 /* selectieve detectie ingang index ds48i */
#define kwc49i        57 /* selectieve detectie ingang index ds49i */
#define kwc02u        58 /* selectieve detectie uitgang index ds02u */
#define kwc48u        59 /* selectieve detectie uitgang index ds48u */
#define kwc49u        60 /* selectieve detectie uitgang index ds49u */
#define kwcnrvcin     61 /* aantal selectieve ingangen            */
#define kwcnrvcuit    62 /* aantal selectieve uitgangen           */
#define kwcbitype     63 /* bijzondere ingreep type               */
#define kwcbitijd     64 /* bijzondere ingreep tijdsduur         */
#define kwcbifc       65 /* bijzondere ingreep fasecyclus        */
#define kwc01         66 /* fasecyclus afhankelijke gegevens fc01 */
#define kwc02         67 /* fasecyclus afhankelijke gegevens fc02 */
#define kwc03         68 /* fasecyclus afhankelijke gegevens fc03 */
#define kwc05         69 /* fasecyclus afhankelijke gegevens fc05 */
#define kwc08         70 /* fasecyclus afhankelijke gegevens fc08 */
#define kwc09         71 /* fasecyclus afhankelijke gegevens fc09 */
#define kwc10         72 /* fasecyclus afhankelijke gegevens fc10 */
#define kwc11         73 /* fasecyclus afhankelijke gegevens fc11 */
#define kwc22         74 /* fasecyclus afhankelijke gegevens fc22 */
#define kwc26         75 /* fasecyclus afhankelijke gegevens fc26 */
#define kwc28         76 /* fasecyclus afhankelijke gegevens fc28 */
#define kwc48         77 /* fasecyclus afhankelijke gegevens fc48 */
#define kwc49         78 /* fasecyclus afhankelijke gegevens fc49 */
#define kwcversie     79 /* programmeervoorschrift versie        */
#define kwcid         80 /* programmeervoorschrift identificer    */

#define PRMMAX       81 /* aantal parameters                    */
```

6.1.2 Instellingen

```
PRM(kwcnrklas , "KWCNRKLAS" , 0 , RO_type ); /* aantal klassificatie ingangen          */
PRM(kwcnrlength, "KWCNRLENGTH", 0 , RO_type ); /* aantal lengtedetectie 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 */
PRM(kwc02u , "KWC02U" , ds02u , RO_type ); /* selectieve detectie uitgang index ds02u */
PRM(kwc48u , "KWC48U" , ds48u , RO_type ); /* selectieve detectie uitgang index ds48u */
```

Datum	Versie	Status	Documentnaam	Auteur
28-01-09	005		Programmeer Voorschrift IT&T Kwaliteitscentrale 3	RKa

```

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(kwcversie , "KWCVERSIE" , 3 , RO_type ); /* programmeervoorschrift versie */
PRM(kwcid , "KWCID" , 32493 , RO_type ); /* programmeervoorschrift identifier */

```

6.1.3 Regel bestand

Het hier opgenomen VAG voorbeeld past bij de manier van implementatie van de IVER afhandeling in gebruik bij IT&T. Dit voorbeeld is op dit punt derhalve niet gegarandeerd toepasbaar indien afhandeling op een andere manier geschiedt.

```

#ifndef AUTOMAAT
#include "mv_ccol.h"
#else
#include "mv_ccol.c"
#endif

void KwcSetVagBits(count fc, mulv vag, count prm)
{
if (CG[fc] > CG_FG && CG[fc] < CG_GL && vag < 5) {
if (WG[fc]) {
if (vag <= 2) { /* stap over WG heen */
PRM[prm] |= ((vag&0x07) << 4);
}
}
else if (MG[fc]) {
if (!(YM[fc]&~BIT3)) {
PRM[prm] |= ((vag&0x07) << 4);
}
}
}
}

```

```
    }
    else {
        PRM[prm] |= ((vag&0x07) << 4);
    }
}
}
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] |= 0x0004; /* vecom prioriteits realisatie */
    if (SH[hov48]) PRM[kwc48] |= 0x0004; /* vecom prioriteits realisatie */
    if (SH[hov49]) PRM[kwc49] |= 0x0004; /* vecom prioriteits realisatie */


    /* ov in/uitmeldingen (geen DS/KAR-modude) */
    if (SD[ds02i]) PRM[kwc02] |= 0x0200; /* ov inmelding */
    if (SD[ds02u]) PRM[kwc02] |= 0x0400; /* ov uitmelding */

    /* VAG doorgifte */
    KwcSetVagBits(fc02, MM[mvag02], kwc02);

    /* 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; /* fixatie */
        PRM[kwcbitijd] = CIF_KLOK[CIF_SEC_TELLER] - klok_fix;
        klok_fix = 0;
    }
}

void system_application(void)
```

	Datum 28-01-09	Versie 005	Status	Documentnaam Programmeer Voorschrift IT&T Kwaliteitscentrale 3	Auteur RKa
---	-------------------	---------------	--------	--	---------------

```
{  
    ...  
    KwcApplication();  
    MvSave();  
}
```