



Programmeer Voorschrift IT&T Kwaliteitscentrale

(VERSIE 2V)
(VIALIS VARIANT)

	Datum	Versie	Status	Documentnaam:	Auteur :
	31-05-11	005	AC	PROGRAMMEER VOORSCHRIFT IT&T KWALITEITSCENTRALE 2V.DOC	RKa
				CC:	


DOCUMENT HISTORIE

wijziging door:	datum:	omschrijving:
RKa	16-10-2001	
RKa	05-09-2003	Document apart genomen uit MV-module interface doc.
RKa	13-11-2003	Kopie van versie 2, aangepast voor Vialis situatie

	Datum	Versie	Status	Documentnaam:	Auteur :
	31-05-11	005	AC	PROGRAMMEER VOORSCHRIFT IT&T KWALITEITSCENTRALE 2V.DOC	RKa
				CC:	

INHOUD


1	Inleiding	4
2	Programmeer voorschrift.....	5
2.1	CCOL	5
2.2	RWS-C	5
2.3	Applicatie specifieke parameters	6
2.3.1	Definities	6
2.3.2	Parameter opbouw	7
3	BIJLAGEN.....	10
3.1	Voorbeeld	10

	Datum 31-05-11	Versie 005	Status AC	Documentnaam: PROGRAMMEER VOORSCHRIFT IT&T KWALITEITSCENTRALE 2V.DOC	Auteur : RKa
				CC:	

1 Inleiding

Dit document is van toepassing voor de aanmaak van MV-files in een Vialis regelaar. Bij een Vialis regelaar worden de standaard MV-events geproduceerd door de procesbesturing.

In dit document wordt beschreven welke extra gegevens een applicatie kan leveren via de CVN-interface, t.b.v. het genereren van *extra* events, die zonder deze toevoegingen niet zijn te bepalen.

	Datum	Versie	Status	Documentnaam:	Auteur :
	31-05-11	005	AC	PROGRAMMEER VOORSCHRIFT IT&T KWALITEITSCENTRALE 2V.DOC	RKa
				CC:	

2 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.

2.1 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.

Om de MV-module te gebruiken in een **simulatie omgeving** 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. In het project dient mv_ccol.obj en mv_fab.obj te worden opgenomen.

2.2 RWS-C

Om de MV-module te gebruiken in een **simulatie omgeving** dient in de RWS-C functie *overige_voorwaarden()* een call naar de functie *MvSave()* gezet te worden. In hetzelfde applicatiebestand (crapcod.c) dient de header "mv-rwsc.c" geinclude te worden. In het project dient mv_fab.obj te worden opgenomen.

Om in een *simulatie omgeving* toch de beschikking te hebben over logische namen voor alle IO van de regeling dient er voor iedere applicatie een "naamfile" aangemaakt te worden met de naam "mvf.dat". In deze file dienen de namen te staan in volgorde van de cvn interface. De file wordt runtime geraadpleegd door de applicatie in de simulatie omgeving.

Opbouw van de file "mvf.dat":


```
[SYSTEM]
<system_name>      = Naam van de kruising (invoer naam voor de kwaliteits centrale)

[SIGNALGROUPS]
<SG01>             = Namen van de signaalgroepen
...
<SGnn>

[OUTPUTS]
<output_name_1>    = Logische namen voor de outputs (koppelsignalen en overige outputs)
...
<output_name_n>

[DETECTORS]
<SG01_1>           = Namen van de detectoren
...
<SGnn_x>

[INPUTS]
<input_name_1>     = Logische namen voor de inputs (koppelsignalen en overige inputs)
...
<input_name_n>
```

	Datum	Versie	Status	Documentnaam:	Auteur :
	31-05-11	005	AC	PROGRAMMEER VOORSCHRIFT IT&T KWALITEITSCENTRALE 2V.DOC	RKa
				CC:	

Een voorbeeld file:

```

[SYSTEM]
KP106

[SIGNALGROUPS]
SG02
SG05
SG08
SG11

[OUTPUTS]
BUS_OP_SG02
BUS_OP_SG08
BUS_OP_SG11

[DETECTORS]
SG02_1
SG02_2
SG05_1
SG05_2
SG05_3
SG05_4
SG08_1
SG08_2
SG11_1
SG11_2
SG08_3
SG08_4
SG11_3
SG11_4

[INPUTS]

```

2.3 Applicatie specifieke parameters

2.3.1 Definities

Prioriteits ingreep (zie Parameter opbouw: **6**)

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.

Bijzondere ingreep (zie Parameter opbouw: **5**)

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.

Snelheids meting (zie Parameter opbouw: **7**)

De snelheid zoals die door de bijbehorende detector is gemeten.


Max groen bereikt (zie Parameter opbouw: **4**)

Een regeling heeft maxgroen bereikt indien verlengroen (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, retour wachtgroen).

Module afloop start (zie Parameter opbouw: **3**)

Het moment dat de kruispuntscyclus opnieuw start.

Alternatieve groen realisatie (zie Parameter opbouw: **4**)

	Datum	Versie	Status	Documentnaam:	Auteur :
	31-05-11	005	AC	PROGRAMMEER VOORSCHRIFT IT&T KWALITEITSCENTRALE 2V.DOC	RKa
				CC:	

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

2.3.2 Parameter opbouw

Alle applicatie afhankelijke informatie nodig voor de Kwaliteitscentrale wordt aangeboden via vrije parameters op de CVN-interface (PRM voor CCOL, EPARMS 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 afhankelijk van het aantal selectieve in- en uitgangen en van het aantal snelheids detectie lussen. 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 \Rightarrow **CIF_PARM1[CIF_AANT_PARM1-1] = 32493**

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

Resultaat \Rightarrow **CIF_PARM1[CIF_AANT_PARM1-2] = 2**

3. De parameter hiervoor geeft het actuele blok/module nummer. Een regeling volgens een langstwachende structuur dient op deze plaats een 0 op te geven.

Resultaat \Rightarrow **CIF_PARM1[CIF_AANT_PARM1-3] = actieve module/blok of 0**

4. 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:

Bit 0: de richting heeft maxgroen bereikt.

Bit 1: de richting is alternatief gerealiseerd.

Bit 2: de richting kent een prioriteits ingreep.

Bit 3: start aanvraag op de richting

Bit 15: schakelaar bit: GEEN mv berichten 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.

Resultaat \Rightarrow bij FCMAX = n fasecycli ($fc_0 \dots fc_{n-1}$)

CIF_PARM1[CIF_AANT_PARM1 - 3 - FCMAX] = gegevens eerste fasecyclus (fc_0)

CIF_PARM1[CIF_AANT_PARM1 - 3 - FCMAX+1] = gegevens tweede fasecyclus (fc_1)

...

CIF_PARM1[CIF_AANT_PARM1 - 3 - FCMAX+(n-1)] = gegevens n-de fasecyclus (fc_{n-1})

5. Voor bijzondere ingrepen worden 2 parameters gereserveerd

De parameters dienen alle in 1 keer gevuld te worden bij het einde van een bijzondere ingreep. In de eerste parameter dient voor het vastleggen van het type ingreep.

1 = AHOB


2 = Brug

3 = File

4 = HV

5 = overig

De tweede parameter geeft de tijdsduur aan dat de betreffende ingreep geduurd heeft.

	Datum	Versie	Status	Documentnaam:	Auteur :
	31-05-11	005	AC	PROGRAMMEER VOORSCHRIFT IT&T KWALITEITSCENTRALE 2V.DOC	RKa
				CC:	

Resultaat ⇒

$CIF_PARAM1[CIF_AANT_PARAM1 - 3 - FCMAX - 2] = \text{ingreep type}$
 $CIF_PARAM1[CIF_AANT_PARAM1 - 3 - FCMAX - 1] = \text{tijdsduur}$

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

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

Hiervoor worden er per selectief ingangssignaal en per selectief uitgangssignaal 2 parameters gereserveerd. Eén t.b.v. opgeven locatie in IS buffer waar de betreffende ingang zich bevindt en één om indien er een melding op de selectieve ingang plaatsvindt het lijnnummer van de inmelder door te kunnen geven.

De parameters zijn als volgt gegroepeerd: index parameters selectieve inmelding ingang, index parameters selectieve uitmelding ingang, lijnnummer parameters selectieve ingangen en lijnnummer parameters selectieve uitgangen, parameter aantal selectieve ingangen, parameter aantal selectieve uitgangen.

Resultaat ⇒ bij p selectieve ingangen en q selectieve uitgangen

$CIF_PARAM1[CIF_AANT_PARAM1 - 7 - FCMAX - q - p - q - p] = \text{index } 1^{\text{e}} \text{ sel. ingang}$

...

$CIF_PARAM1[CIF_AANT_PARAM1 - 7 - FCMAX - q - p - q - 2] = \text{index } p^{\text{e}} - 1 \text{ sel. ingang}$

$CIF_PARAM1[CIF_AANT_PARAM1 - 7 - FCMAX - q - p - q - 1] = \text{index } p^{\text{e}} \text{ sel. ingang}$

$CIF_PARAM1[CIF_AANT_PARAM1 - 7 - FCMAX - q - p - q] = \text{index } 1^{\text{e}} \text{ sel. uitgang}$

...

$CIF_PARAM1[CIF_AANT_PARAM1 - 7 - FCMAX - q - p - 2] = \text{index } q^{\text{e}} - 1 \text{ sel. uitgang}$

$CIF_PARAM1[CIF_AANT_PARAM1 - 7 - FCMAX - q - p - 1] = \text{index } q^{\text{e}} \text{ sel. uitgang}$

$CIF_PARAM1[CIF_AANT_PARAM1 - 7 - FCMAX - q - p] = \text{lijnnummer } 1^{\text{e}} \text{ sel. ingang}$

$CIF_PARAM1[CIF_AANT_PARAM1 - 7 - FCMAX - q - 2] = \text{lijnnummer } p^{\text{e}} - 1 \text{ sel. ingang}$

$CIF_PARAM1[CIF_AANT_PARAM1 - 7 - FCMAX - q - 1] = \text{lijnnummer } p^{\text{e}} \text{ sel. ingang}$

$CIF_PARAM1[CIF_AANT_PARAM1 - 7 - FCMAX - q] = \text{lijnnummer } 1^{\text{e}} \text{ sel. uitgang}$

...

$CIF_PARAM1[CIF_AANT_PARAM1 - 7 - FCMAX - 2] = \text{lijnnummer } q^{\text{e}} - 1 \text{ sel. uitgang}$

$CIF_PARAM1[CIF_AANT_PARAM1 - 7 - FCMAX - 1] = \text{lijnnummer } q^{\text{e}} \text{ sel. uitgang}$

$CIF_PARAM1[CIF_AANT_PARAM1 - 5 - FCMAX - 2] = p$

$CIF_PARAM1[CIF_AANT_PARAM1 - 5 - FCMAX - 1] = q$

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

De eerste parameter voor de parameters uit punt 7 geeft aan hoeveel snelheids detectie meldingen er in de applicatie gebruikt worden. Voor ieder snelheids detectie melding worden twee parameters gedefinieerd. Eén t.b.v. opgeven locatie in IS buffer waar de betreffende ingang zich bevindt en één om de gemeten snelheid van de betreffende ingang door te geven aan de MV-module.

De parameters zijn als volgt gegroepeerd: index parameters snelheids detectie ingangen, snelheids melding parameters, parameter aantal snelheids detectie ingangen.

Resultaat ⇒ bij p selectieve ingangen en q selectieve uitgangen en r snelheids detectie ingangen

$CIF_PARAM1[CIF_AANT_PARAM1 - 8 - FCMAX - 2*p - 2*q - r - r] = \text{idx } 1^{\text{e}} \text{ detector}$

...

$CIF_PARAM1[CIF_AANT_PARAM1 - 8 - FCMAX - 2*p - 2*q - r - 2] = \text{idx } r^{\text{e}} - 1 \text{ detector}$

$CIF_PARAM1[CIF_AANT_PARAM1 - 8 - FCMAX - 2*p - 2*q - r - 1] = \text{idx } r^{\text{e}} \text{ detector}$


$CIF_PARAM1[CIF_AANT_PARAM1 - 8 - FCMAX - 2*p - 2*q - r] = v \text{ van } 1^{\text{e}} \text{ detector}$


...

$CIF_PARAM1[CIF_AANT_PARAM1 - 8 - FCMAX - 2*p - 2*q - 2] = v \text{ van } r^{\text{e}} - 1 \text{ detector}$

$CIF_PARAM1[CIF_AANT_PARAM1 - 8 - FCMAX - 2*p - 2*q - 1] = v \text{ van } r^{\text{e}} \text{ snelheids detector}$

$CIF_PARAM1[CIF_AANT_PARAM1 - 7 - FCMAX - 2*p - 2*q - 1] = r$

	Datum 31-05-11	Versie 005	Status AC	Documentnaam: PROGRAMMEER VOORSCHRIFT IT&T KWALITEITSCENTRALE 2V.DOC	Auteur : RKa
				CC:	

	Datum	Versie	Status	Documentnaam:	Auteur :
	31-05-11	005	AC	PROGRAMMEER VOORSCHRIFT IT&T KWALITEITSCENTRALE 2V.DOC	RKa
				CC:	

3 BIJLAGEN

3.1 Voorbeeld


Definitie bestand

```

#define prmkwcnrsnel          322 /* aantal snelheidsdetectie ingangen          */
#define prmkwcs022           323 /* selectieve detectie ingang index ds022      */
#define prmkwcs082           324 /* selectieve detectie ingang index ds082      */
#define prmkwcs021           325 /* selectieve detectie uitgang index ds021     */
#define prmkwcs081           326 /* selectieve detectie uitgang index ds081     */
#define prmkwclijs022        327 /* lijnummer ds022                            */
#define prmkwclijs082        328 /* lijnummer ds082                            */
#define prmkwclijs021        329 /* lijnummer ds021                            */
#define prmkwclijs081        330 /* lijnummer ds081                            */
#define prmkwcnrvcin         331 /* aantal selectieve ingangen                  */
#define prmkwcnrvcuit        332 /* aantal selectieve uitgangen                 */
#define prmkwcbitype         333 /* bijzondere ingreep type                     */
#define prmkwcbitijd         334 /* bijzondere ingreep tijdsduur                */
#define prmkwc01             335 /* fasecyclus afhankelijke gegevens fc01      */
#define prmkwc02             336 /* fasecyclus afhankelijke gegevens fc02      */
#define prmkwc03             337 /* fasecyclus afhankelijke gegevens fc03      */
#define prmkwc05             338 /* fasecyclus afhankelijke gegevens fc05      */
#define prmkwc07             339 /* fasecyclus afhankelijke gegevens fc07      */
#define prmkwc08             340 /* fasecyclus afhankelijke gegevens fc08      */
#define prmkwc09             341 /* fasecyclus afhankelijke gegevens fc09      */
#define prmkwc105            342 /* fasecyclus afhankelijke gegevens fc105     */
#define prmkwc108            343 /* fasecyclus afhankelijke gegevens fc108     */
#define prmkwc110            344 /* fasecyclus afhankelijke gegevens fc110     */
#define prmkwc111            345 /* fasecyclus afhankelijke gegevens fc111     */
#define prmkwc112            346 /* fasecyclus afhankelijke gegevens fc112     */
#define prmkwc126            347 /* fasecyclus afhankelijke gegevens fc126     */
#define prmkwc136            348 /* fasecyclus afhankelijke gegevens fc136     */
#define prmkwc22             349 /* fasecyclus afhankelijke gegevens fc22      */
#define prmkwc24             350 /* fasecyclus afhankelijke gegevens fc24      */
#define prmkwc26             351 /* fasecyclus afhankelijke gegevens fc26      */
#define prmkwc27             352 /* fasecyclus afhankelijke gegevens fc27      */
#define prmkwc28             353 /* fasecyclus afhankelijke gegevens fc28      */
#define prmkwc31             354 /* fasecyclus afhankelijke gegevens fc31      */
#define prmkwc32             355 /* fasecyclus afhankelijke gegevens fc32      */
#define prmkwc33             356 /* fasecyclus afhankelijke gegevens fc33      */
#define prmkwc34             357 /* fasecyclus afhankelijke gegevens fc34      */
#define prmkwc35             358 /* fasecyclus afhankelijke gegevens fc35      */
#define prmkwc36             359 /* fasecyclus afhankelijke gegevens fc36      */
#define prmkwc37             360 /* fasecyclus afhankelijke gegevens fc37      */
#define prmkwc38             361 /* fasecyclus afhankelijke gegevens fc38      */
#define prmkwcblok           362 /* actuele blok                                */
#define prmkwcversie         363 /* programmeervoorschrift versie               */
#define prmkwcid             364 /* programmeervoorschrift identificier         */

#define PRMMAX                365 /* aantal parameters                           */

```


	Datum	Versie	Status	Documentnaam:	Auteur :
	31-05-11	005	AC	PROGRAMMEER VOORSCHRIFT IT&T KWALITEITSCENTRALE 2V.DOC	RKa
				CC:	

Instellingen

```

PRM(prmkwcnrsnel , "KWCNRSNEL" , 0, RO_type); /* aantal snelheidsdetectie ingangen */
PRM(prmkwcs022 , "KWCS022" , ds022, RO_type); /* selectieve detectie ingang index ds022 */
PRM(prmkwcs082 , "KWCS082" , ds082, RO_type); /* selectieve detectie ingang index ds082 */
PRM(prmkwcs021 , "KWCS021" , ds021, RO_type); /* selectieve detectie uitgang index ds021 */
PRM(prmkwcs081 , "KWCS081" , ds081, RO_type); /* selectieve detectie uitgang index ds081 */
PRM(prmkwclijs022, "KWCLIJNS022", 0, 0 ); /* lijnummer ds022 */
PRM(prmkwclijs082, "KWCLIJNS082", 0, 0 ); /* lijnummer ds082 */
PRM(prmkwclijs021, "KWCLIJNS021", 0, 0 ); /* lijnummer ds021 */
PRM(prmkwclijs081, "KWCLIJNS081", 0, 0 ); /* lijnummer ds081 */
PRM(prmkwcnrvcin , "KWCNRVCIN" , 2, RO_type); /* aantal selectieve ingangen */
PRM(prmkwcnrvcuit , "KWCNRVCUIT" , 2, RO_type); /* aantal selectieve uitgangen */
PRM(prmkwcbitype , "KWCBITYPE" , 0, 0 ); /* bijzondere ingreep type */
PRM(prmkwcbitijd , "KWCBITIJD" , 0, 0 ); /* bijzondere ingreep tijdsduur */
PRM(prmkwc01 , "KWC01" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc01 */
PRM(prmkwc02 , "KWC02" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc02 */
PRM(prmkwc03 , "KWC03" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc03 */
PRM(prmkwc05 , "KWC05" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc05 */
PRM(prmkwc07 , "KWC07" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc07 */
PRM(prmkwc08 , "KWC08" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc08 */
PRM(prmkwc09 , "KWC09" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc09 */
PRM(prmkwc105 , "KWC105" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc105 */
PRM(prmkwc108 , "KWC108" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc108 */
PRM(prmkwc110 , "KWC110" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc110 */
PRM(prmkwc111 , "KWC111" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc111 */
PRM(prmkwc112 , "KWC112" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc112 */
PRM(prmkwc126 , "KWC126" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc126 */
PRM(prmkwc136 , "KWC136" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc136 */
PRM(prmkwc22 , "KWC22" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc22 */
PRM(prmkwc24 , "KWC24" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc24 */
PRM(prmkwc26 , "KWC26" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc26 */
PRM(prmkwc27 , "KWC27" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc27 */
PRM(prmkwc28 , "KWC28" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc28 */
PRM(prmkwc31 , "KWC31" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc31 */
PRM(prmkwc32 , "KWC32" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc32 */
PRM(prmkwc33 , "KWC33" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc33 */
PRM(prmkwc34 , "KWC34" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc34 */
PRM(prmkwc35 , "KWC35" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc35 */
PRM(prmkwc36 , "KWC36" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc36 */
PRM(prmkwc37 , "KWC37" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc37 */
PRM(prmkwc38 , "KWC38" , 0, 0 ); /* fasecyclus afhankelijke gegevens fc38 */
PRM(prmkwcblok , "KWCBLOK" , 0, 0 ); /* actuele blok */
PRM(prmkwcversie , "KWCVERSIE" , 2, RO_type); /* programmeervoorschrift versie */
PRM(prmkwcid , "KWCID" , 32493, RO_type); /* programmeervoorschrift identifier */

```

	Datum	Versie	Status	Documentnaam:	Auteur :
	31-05-11	005	AC	PROGRAMMEER VOORSCHRIFT IT&T KWALITEITSCENTRALE 2V.DOC	RKa
				CC:	

Regel bestand

```

/* globale data */
/* ----- */
bool a[FCMAX];          /* t.b.v. start aanvraag vaststelling voor kwc */

#ifdef AUTOMAAT
#include "mv_ccol.h"
#endif

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

    /* fasecyclus parameters */
    /* ----- */
    for (i = prmkwc01; i <= prmkwc38; ++i)
        PRM[i] &= 0x8000; /* reset alle bits (uitgezonderd schakelaar bit) */

    /* richtingen met prioriteits realisatie */
    if (SH[hvc02]) PRM[prmkwc02] |= 0x04; /* vecom prioriteits realisatie */
    if (SH[hvc08]) PRM[prmkwc08] |= 0x04; /* vecom prioriteits realisatie */

    for (i=0; i<FCMAX; ++i) {
        if (EG[i] && AR[i]) {
            PRM[prmkwc01+i] |= 0x02; /* alternatieve realisatie */
        }
        if (A[i] != a[i]) {
            if (A[i]) {
                PRM[prmkwc01+i] |= 0x08; /* start aanvraag */
            }
            a[i] = A[i];
        }
    }

    /* start module */
    /* ----- */
    PRM[prmkwcblok] = 0;

    if (SML) {
        PRM[prmkwcblok] = (ML+1);
    }

    /* bijzondere ingrepen */
    /* ----- */
    PRM[prmkwcbitijd] = 0;
    PRM[prmkwcbitype] = 0;

    if (CIF_IS[isfix] && (klok_fix == 0))
        klok_fix = CIF_KLOK[CIF_SEC_TELLER];
    else if ((klok_fix != 0) && !CIF_IS[isfix]) {
        PRM[prmkwcbitype] = 5; /* overig */
        PRM[prmkwcbitijd] = CIF_KLOK[CIF_SEC_TELLER] - klok_fix;
        klok_fix = 0;
    }
}

void application(void)
{
    ...
    KwcApplication();
}

void system_application(void)
{
    ...
#ifdef AUTOMAAT
    MvSave();
#endif
}

```