

De CVN C-interface

Beschrijving van
de software-interface tussen
het applicatieprogramma en
de procesbesturing
voor verkeersregeltoestellen

CVN Commissie C / Ton van Grinsven en Marcel Fick
versienummer: 4.0
datum: 12 oktober 2011
status: definitief

INHOUD

1	INLEIDING	3
2	Globale functionele indeling van het regeltoestel	6
3	APPLICATIEPROGRAMMA	7
3.1	Algemeen.....	7
3.2	Besturing applicatieprogramma.....	7
4	PROCESBESTURING	8
4.1	Algemeen.....	8
4.2	Reactietijd.....	8
4.3	Voorwaarden aansturen lantaarns door de procesbesturing	8
4.3.1	Afhandeling volgorde en conflicten door de procesbesturing.....	9
4.3.2	Afhandeling garantietijden door de procesbesturing.....	9
5	INTERFACE.....	10
5.1	Algemeen.....	10
5.1.1	Interface opbouw.....	10
5.1.2	Variabelen / woordlengte	12
5.1.3	Codering.....	12
5.1.4	Initialisatie.....	12
5.1.5	Lezen en schrijven in de interface	13
5.1.6	Gereserveerd.....	13
5.2	In- en uitgangssignalen van het applicatieprogramma	13
5.2.1	Gewenste uitgangssturing.....	14
5.2.2	Werkelijke uitgangssturing.....	14
5.2.3	Ingangssignalen	15
5.3	Programmastatus	15
5.3.1	Gewenste programmastatus.....	16
5.3.2	Werkelijke programmastatus.....	16
5.4	Berichten met uitzondering van berichten voor monitoring.....	17
5.4.1	Uitgaande berichten van het applicatieprogramma	17
5.4.2	Ingaande berichten voor het applicatieprogramma	18
5.5	Berichten ten behoeve van monitoring.....	18
5.5.1	Uitgaande berichten van het applicatieprogramma t.b.v. verkeersmonitoring.....	18
5.5.2	Ingaande berichten voor het applicatieprogramma t.b.v. verkeersmonitoring	19
5.6	Berichten t.b.v. fileopslag van verkeersgegevens.....	19
5.6.1	Uitgaande berichten van het applicatieprogramma t.b.v. fileopslag	20
5.6.2	Filewisselpunt.....	20
5.6.3	File-extensie	21
5.7	Systeemklok	21
5.8	Parameters	22
5.9	Selectieve detectie	22
5.10	Lezen en schrijven in de interface	24
5.11	Ontruimingstijden.....	26
6	SLOTOPMERKINGEN.....	28
6.1	Realisatie van de C-interface.....	28
6.2	Reactietijd.....	28
6.3	Nader te regelen.....	28
7	VERKLARING GEBRUIKTE SYMBOLEN	29
8	IMPLEMENTATIE.....	31

1 INLEIDING

De gebruikers van verkeersregeltoestellen, vertegenwoordigd in de CVN, zijn van mening dat het van het grootste belang is te komen tot een fabrikant-onafhankelijke specificatietaal.

Het doel hiervan is dat de gebruikers die taal kunnen hanteren voor het eenduidig opstellen van verkeersregeltechnische specificaties.

Deze specificaties moeten dan zonder interpretatieverschillen in het regeltoestel kunnen worden geïmplementeerd.

In het verleden is er al eens zo'n taal gecreëerd (TRAFCOL) die de voor de gebruiker noodzakelijke eigenschappen bezit, doch het voordeel hiervan is nooit helemaal uit de verf gekomen vanwege het ontbreken van de programmatuur die het handmatig vertalen en daarmee de interpretatieverschillen uitbande.

Door de ontwikkelingen in het recente verleden op het gebied van de microcomputer (PC), de verkeersregeltoestellen, de daarbij behorende software, het beschikbaar komen van relatief goedkope compilers en niet in het minst het in Delft geleverde bewijs dat het mogelijk is, lijkt de tijd rijp een hernieuwde poging te wagen te komen tot de definitie van een, aan de stand van de techniek aangepaste, fabrikantonafhankelijke waarde vrije specificatietaal.

De CVN heeft zich tot taak gesteld in deze ontwikkeling een trekkersrol te vervullen en heeft hiertoe een werkgroep ingesteld die als opdracht heeft het opstellen van een voorstel voor een beschrijving van de benodigde interface tussen het gedeelte dat door de gebruiker zal worden toegeleverd (applicatieprogramma) en het gedeelte dat door de fabrikant moet worden verzorgd (procesbesturing). Op basis van dit voorstel zullen gesprekken met de fabrikanten worden gevoerd om te komen tot een daadwerkelijke realisering van de door de gebruikers gewenste situatie.

Bij het opstellen van de beschrijving van de interface zijn de volgende stappen doorlopen:

- het opstellen van een globale functionele indeling van een regeltoestel
- het opstellen van een functionele beschrijving van het gebruikersdeel (applicatieprogramma)
- het opstellen van een functionele beschrijving van het fabrikantdeel (procesbesturing)
- het opstellen van een beschrijving van de interface tussen de procesbesturing en het applicatieprogramma.

Gezien de ervaring in het verleden dient de specificatietaal een bestaande, gestandaardiseerde, hogere programmeertaal te zijn, waardoor gebruik gemaakt kan worden van compilers die getoetst zijn aan de taalstandaard.

Op basis van de voornoemde overwegingen heeft de werkgroep besloten de keuze te bepalen op de taal "C" (volgens de ANSI standaard).

Commissie-C van de CVN

A. Beckers
A. van Grinsven
P. de Jong
W. Kooij
C. Moons
Th. Muller
P. van Riel (tot februari 1990)

12 februari 1990

Bij de gewijzigde versie van 26 juni 1992

Sinds de uitgave van dit document op 12 februari 1990 zijn een aantal opdrachtgevers daadwerkelijk aan de slag gegaan met het schrijven van applicatieprogramma's in C, e.e.a. gebaseerd op de in dit document vastgelegde interface. Een aantal van deze applicatieprogramma's zijn succesvol geïmplementeerd in regeltoestellen en draaien inmiddels op straat.

Dit had tot direct gevolg dat de interface in de praktijk kon worden getest en geëvalueerd op juistheid en bruikbaarheid. Uit deze evaluatie bleek dat de interfacebeschrijving op een aantal plaatsen kon worden verbeterd.

De Commissie-C van de CVN is daartoe in het voorjaar van 1992 opnieuw bij elkaar gekomen om voorstellen voor verbeteringen te formuleren. Over deze voorstellen is bijzonder constructief overleg geweest met alle betrokken fabrikanten, hetgeen heeft geleid tot de voorliggende definitieve beschrijving van de software-interface tussen het applicatieprogramma en de procesbesturing voor verkeersregeltoestellen.

Bij de versie 31

Anno 2004 kan worden gesteld dat de C-interface zeer breed wordt gebruikt in Nederland. Ontwikkelen van fabrikantonafhankelijke applicatieprogramma's door of in opdracht van de wegbeheerder is gemeengoed geworden.

De ontwikkelingen in het vakgebied zijn echter niet stil blijven staan. Daarom leek het de schrijvers van versie 3 goed om een nieuwe versie van dit document uit te brengen.

Uitgangspunt daarbij is geweest de oorspronkelijke interface zoveel mogelijk intact te laten, enkele nieuwe aspecten daaraan toe te voegen en voor de toekomst enige ruimte tot flexibiliteit te bieden.

De CVN, als oorspronkelijke initiatiefnemer van de C-interface, heeft haar instemming met deze nieuwe versie betuigd.

Deze nieuwe versie is eveneens aan de ASTRIN voorgelegd om commentaar. Na doorvoering van enige modificaties, waarna de versie 3.1 ontstond, heeft de ASTRIN geen bezwaar tegen invoering van deze versie.

Ton van Grinsven
Peter de Jong
Frank Peters
mei 2005

Bij de versie 32

Bij de implementatie van de C-interface is een klein foutje aan het licht gekomen in de cif.inc bij de declaratie van CIF_DSI[CIF_AANT_DSI]. Dit is in hoofdstuk 8 van het document aangepast.

Martin Barto
i.o.m. Ton van Grinsven en Frank Peters
mei 2006

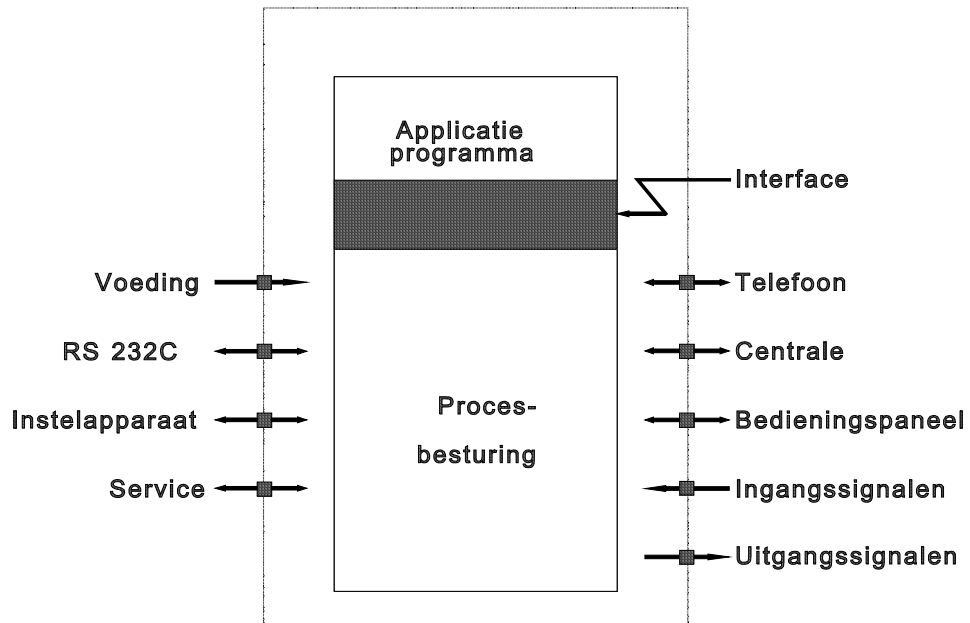
Bij de versie 40

Voor het loggen van verkeersgegevens vanuit het applicatieprogramma t.b.v. fileopslag van deze verkeersgegevens in het verkeersregeltoestel is een apart communicatiekanaal nodig dat geschikt is voor binaire data-overdracht. Hiertoe is de CVN C-interface in overleg met de ASTRIN werkgroep KAR uitgebreid met een extra berichtenbuffer.

Ton van Grinsven
Marcel Fick
oktober 2011

2 GLOBALE FUNCTIONELE INDELING VAN HET REGELTOESTEL

In onderstaand figuur is een functionele verdeling van een regeltoestel met z'n mogelijke verbindingen naar de buitenwereld weergegeven.



figuur 1

Op basis van deze verdeling zullen de eigenschappen van het applicatieprogramma en de procesbesturing worden gespecificeerd.

De aangegeven RS232c-poort is primair bedoeld voor het benaderen van de interface door de gebruiker voor het lezen en schrijven van berichten (zie paragraaf 5.4.1 en 5.4.2).

3 APPLICATIEPROGRAMMA

3.1 Algemeen

Het applicatieprogramma biedt, aan de hand van de door de procesbesturing op de interface aangeboden informatie, de gewenste uitgangssturing, gewenste status, berichten en parameters aan aan de interface ter afhandeling door de procesbesturing.

Het applicatieprogramma zal door de gebruiker in "C" ontwikkeld worden. Dit bronprogramma moet dan machinaal omgezet worden in een doelprogramma.

3.2 Besturing applicatieprogramma

Het applicatieprogramma wordt bestuurd vanuit een hoofdfunctie *applicatieprogramma()*.

Declaratie: *s_int16 applicatieprogramma (s_int16);*

De procesbesturing dient zolang als mogelijk c.q. gewenst een aanroep te bevatten van de functie *applicatieprogramma()*

Dit houdt tenminste in dat :

- 1 de interface wordt behandeld en
- 2 het applicatieprogramma wordt doorlopen.

Voor wat betreft de frequentie van aanroep van de functie *applicatieprogramma()* dient voldaan te worden aan de specificaties zoals vermeld in hoofdstuk 6.

4 PROCESBESTURING

4.1 Algemeen

De procesbesturing voorziet in alle functies van het verkeersregeltoestel, m.u.v. hetgeen door het applicatieprogramma wordt verzorgd.

De door het applicatieprogramma gewenste uitgangssturing wordt door de procesbesturing tijdens de toestanden 1, 2 en 4 van de norm NEN3384 (verder te noemen de norm), alsmede tijdens centrale sturing, genegeerd tenzij anders is aangegeven.

Het bovenstaande houdt voor de procesbesturing o.a. het volgende in.

- Het doorgeven van alle voor het applicatieprogramma relevante inkomende signalen naar de interface.
- Het realiseren van de gewenste uitgangssturing van de fasecycli t.b.v. de lantaarns. Deze realisatie geschiedt onder voorwaarden van volgorde, garantietijden en ontruimingstijden.
- Het realiseren van de gewenste uitgangssturing van de overige uitgangen.
- Het doorgeven van de boodschappen naar de RS232c-poort.
- Het verzorgen van het in- en uitschakelen volgens de norm.
- Het wijzigen en bijhouden van de parameters.
- Het verzorgen van functies voor tellingen, dichtheidsmetingen, snelheidsmetingen etc. op verzoek van de gebruiker.

De kruispuntafhankelijke gegevens voor de procesbesturing zoals:

- conflictenmatrix
- garantietijden, ontruimingstijden en maximum groentijden
- configuratie van de in- en uitgangssignalen
- etc.

worden per regeltoestel door de gebruiker vastgelegd.

4.2 Reactietijd

Elke gewenste uitgangssturing dient door de procesbesturing zo snel mogelijk als werkelijke uitgangssturing te worden gerealiseerd en te worden teruggemeld op de interface.

Dit geldt alleen voor zover de gewenste sturing niet strijdig is met de afhandeling van de volgorde en/of garantietijden en ontruimingstijden door de procesbesturing.

De werkelijke ingangssignalen op de klemmenstrook van het regeltoestel, danwel op de uitgang van de detectie-unit, dienen binnen 0,1 sec. te worden aangeboden op de interface.

4.3 Voorwaarden aansturen lantaarns door de procesbesturing

De werkelijke uitsturing van de fasecycli dient door de procesbesturing te geschieden met inachtneming van de garantietijden, de ontruimingstijden en de conflicten.

De procesbesturing dient te voorkomen dat een foutieve gewenste uitgangssturing vanuit het applicatieprogramma leidt tot ingrijpen van de bewaking.

Opmerking :
 Groenknipperen wordt daarbij afgehandeld als geel
 Wit/witknipperen wordt daarbij afgehandeld als groen
 Geelknipperen (i.p.v. rood bij voetgangers) wordt daarbij afgehandeld als rood

4.3.1 Afhandeling volgorde en conflicten door de procesbesturing

De signalen worden getoond op basis van de gewenste sturing vanuit het applicatieprogramma, met inachtneming van de in de procesbesturing vastgelegde garantietijden en ontruimingstijden, op de volgende wijze.

- start groen uitwendig treedt op bij: einde rood uitwendig
- einde groen uitwendig treedt op bij: geen groensturing (meer) vanuit de interface en de garantiegroentijd is verstreken

- start geel uitwendig treedt op bij: einde groen uitwendig
- einde geel uitwendig treedt op bij: geen geelsturing (meer) vanuit de interface en de garantiegeeltijd is verstreken
 OF bij:
 maximumgeeltijd is verstreken
 en de garantiegeeltijd is verstreken

- start rood uitwendig treedt op bij: einde geel uitwendig
- einde rood uitwendig treedt op bij: groensturing vanuit de interface en de garantieroodtijd is verstreken en er geen conflicterende groen of geelfasen meer voorkomen
 en geen garantieontruimingstijden meer lopen
 en geen ontruimingstijden meer lopen

In de conflictmatrix ligt vast welke fasecycli onderling conflicterend zijn.

4.3.2 Afhandeling garantietijden door de procesbesturing

Onder garantietijden wordt verstaan de in de procesbesturing vast aangebrachte (=niet wijzigbare) instellingen t.b.v. de betreffende tijdelementen.

- start garantiegroentijd treedt op bij start groen uitwendig
- start garantiegeeltijd treedt op bij start geel uitwendig
- start maximumgeeltijd treedt op bij start geel uitwendig
- start garantieroodtijd treedt op bij start rood uitwendig
- start garantieontruimingstijd en start ontruimingstijd treden op bij einde geel uitwendig

Opmerking:

Intergroentijden mogen niet worden toegepast als vervanging van de garantieontruimingstijden of de ontruimingstijden i.v.m. de mogelijke verlenging van het geel vanuit het applicatieprogramma.

5 INTERFACE

5.1 Algemeen

5.1.1 Interface opbouw

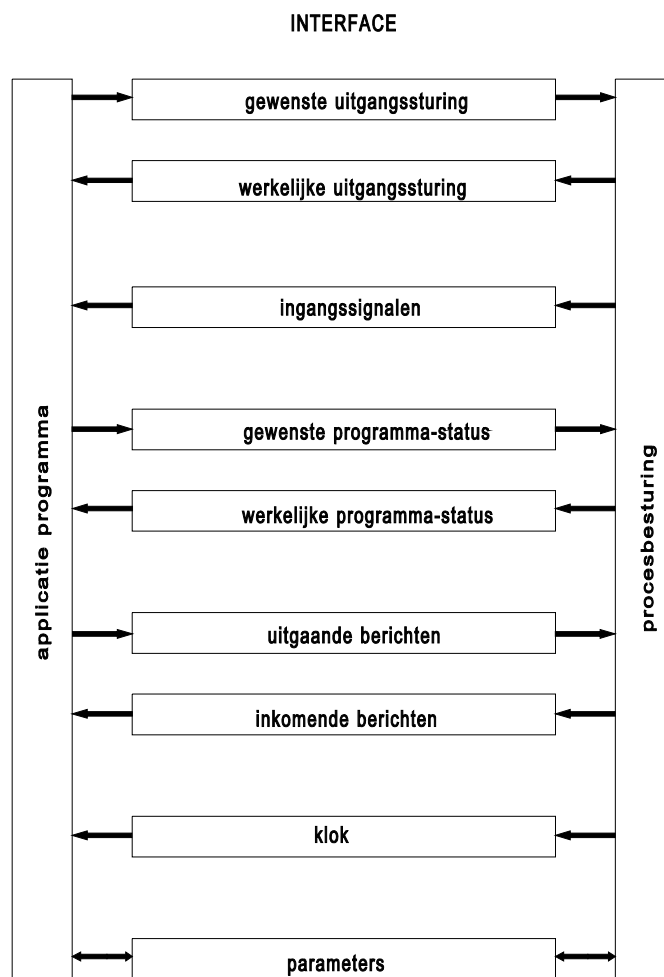
De informatie-uitwisseling tussen het applicatieprogramma en de procesbesturing vindt plaats via de interface.

Deze interface is opgebouwd uit gemeenschappelijke databuffers die, zowel voor het applicatieprogramma als de procesbesturing, bereikbaar zijn. De grootte van de gebruikte buffers wordt per regeltoestel in de interface vastgelegd.

De interface onderscheidt de volgende groepen databuffers:

- in-/uitgangssignalen
- programmastatus
- berichten
- systeemklok
- parameters

In onderstaand figuur zijn deze databuffers schematisch weergegeven.



figuur 2

Bij figuur 2 kan aan de volgende signalen worden gedacht.

gewenste uitgangssturing:

- uitgangsstatus van de fasecyclus (groen/geel/rood)
- brugsignalen
- wachtsignalen voor fietsers (verklikt aanvraag)
- aansturing tekstborden
- waarschuwingslichten

werkelijke uitgangssturing:

- uitgangsstatus van de fasecyclus (groen/geel/rood)
- brugsignalen
- wachtsignalen voor fietsers (verklikt aanvraag)
- aansturing tekstborden
- waarschuwingslichten

ingangssignalen:

- detectie-informatie (detectielus/drukknop)
- selectieve voertuigsignalen
- schakelaar fixatie van het handbedieningspaneel
- schakelaar cyclisch van het handbedieningspaneel
- brugsignalen
- status van de conflictfasen van een fasecyclus

gewenste programmastatus:

- gewenst regelen
- gewenst knipperen

werkelijke programmastatus:

- alles rood status
- werkelijk regelen status
- werkelijk knipperen status
- hardware storing

uitgaande berichten:

- standen van de fasecycli
- monitoringgegevens

inkomende berichten:

- commando's voor het applicatieprogramma

klok:

- jaar
- maand
- dag
- dagsoort (zo t/m za)
- uur
- minuut
- seconde
- secondenteller
- tienden secondenteller

parameters:

- ingestelde waarden tijdelementen
- ingestelde kloktijdstippen
- softwareschakelaars

De databuffers van de interface zullen achtereenvolgens nader worden toegelicht.

5.1.2 Variabelen / woordlengte

Alle variabelen en constanten in de interface hebben, voor betere herkenbaarheid, een naam gekregen die altijd begint met het voorvoegsel “CIF_” (CIF staat voor **C** **I**nter**F**ace).

Voor de variabelen, die in de interface worden vastgelegd, worden aparte typen variabelen gedefinieerd (typedef).

Daar waar een relatie van variabelen binnen het applicatieprogramma met de interface variabelen aanwezig is moeten ook deze variabelen herleid zijn naar de in de interface benoemde type variabelen. De woordlengtes van die variabelen zijn op die manier eenvoudig aan te passen bij implementatie van de applicatie in het regeltoestel.

In dit rapport worden de volgende typen variabelen gebruikt:

- s_int8 8 bits variabele
- s_int16 16 bits variabele
- s_int32 32 bits variabele.

De fabrikant dient bij implementatie er voor zorg te dragen dat bij de typedefine de juiste type variabelen worden gekozen, dat wil zeggen dat de gekozen variabelen minimaal het aantal gespecificeerde bits dienen te bevatten, maar dat vanuit efficiëntie overwegingen (grotere verwerkingssnelheid) voor de types s_int16 en s_int32 een groter type mag worden gekozen.

5.1.3 Codering

In de hierna volgende uitwerking van de databuffers in de interface worden codes met behulp van decimale of binaire waarden als volgt vastgelegd.

- Per codering wordt een voor die variabele unieke decimale waarde bepaald; in dit geval geldt dat de decimale waarde van de variabele relevant is ter bepaling van de betreffende code.
- Per codering wordt voor slechts een of enkele bits een binaire waarde bepaald, voor de overige bits wordt een 'x' getoond (b.v. xxxx xxxx xxxx xx1x); in dit geval geldt dat slechts de waarde van de niet met een 'x' gemarkeerde bits relevant is voor de bepaling van de betreffende code en zijn combinaties toegestaan met waarden van alle overige bits van de variabele.

5.1.4 Initialisatie

Initialisatie van de interface geschiedt door het applicatieprogramma. Voor het initialiseren worden twee verschillende situaties onderscheiden:

1. initialisatie na programmastart en
2. initialisatie voorafgaand aan de toestand regelen.

Ad 1.

Bij initialisatie na programmastart worden door het applicatieprogramma alle variabelen in de interface geïntialiseerd, waarbij van alle parameters de initiële waarde in de interface wordt geschreven. De procesbesturing dient zelf te controleren of in dat geval de waarden voor de parameters worden overgenomen (allereerste maal dat het programma wordt gestart) of dat deze initiële waarden worden genegeerd (en worden overschreven door (gewijzigde) parameterinstellingen uit de procesbesturing (tweede en volgende programmastart)).

Voor de parameter van de functie *applicatieprogramma()* geldt de volgende codering:

<u>decimaal</u>	<u>symbool</u>	<u>betekenis</u>
0	CIF_GEEN_INIT	'normale' aanroep
1	CIF_INIT	initialisatie na programmastart

Voor de returnwaarde van de functie *applicatieprogramma()* geldt de volgende codering:

<u>decimaal</u>	<u>symbool</u>	<u>betekenis</u>
0	CIF_GEEN_FOUT	geen fout opgetreden
-1	CIF_INIT_FOUT	initialisatie fout applicatieprogramma

Door de procesbesturing wordt bij de aanroep van de functie *applicatieprogramma()*

- de waarde CIF_INIT als parameter meegegeven indien geïnitieerd moet worden na start programma.
- de waarde CIF_GEEN_INIT als parameter meegegeven in de overige gevallen van aanroepen van de functie *applicatieprogramma()*.

De functie *applicatieprogramma()* retourneert indien deze functie wordt aangeroepen met de waarde CIF_INIT

- de waarde CIF_GEEN_FOUT bij succesvol initialiseren
- de waarde CIF_INIT_FOUT indien niet succesvol is geïnitieerd; in dat geval mag de functie *applicatieprogramma()* niet meer worden aangeroepen.

De functie *applicatieprogramma()* retourneert indien deze functie wordt aangeroepen met de waarde CIF_GEEN_INIT:

- de waarde CIF_GEEN_FOUT; indien een andere waarde wordt geretourneerd mag de functie *applicatieprogramma()* niet meer worden aangeroepen.

Ad 2

Bij initialisatie voorafgaand aan de toestand regelen worden alleen die variabelen in de interface geïnitieerd die voor het inschakelen van de regeling van belang zijn.

Door het applicatieprogramma wordt aan de hand van de informatie uit de **CIF_WPS[]**-buffer (zie paragraaf 5.3.2) zelfstandig gecontroleerd of er voorafgaand aan de toestand regelen moet worden geïnitieerd.

5.1.5 Lezen en schrijven in de interface

Tijdens uitvoering van de functie *applicatieprogramma()* mag door de procesbesturing niet in de interface worden geschreven of gelezen. De procesbesturing dient ervoor te zorgen dat bij aanroep van de functie *applicatieprogramma()* alle informatie in de interface voor het applicatieprogramma beschikbaar is.

5.1.6 Gereserveerd

Daar waar in dit hoofdstuk wordt gesproken over “gereserveerd” betekent dit:

- dat in de interface ruimte wordt gereserveerd of
- dat de betekenis van de betreffende code is vastgelegd doch de uitwerking daarvan binnen de procesbesturing of het applicatieprogramma niet in dit document is vastgesteld.

5.2 In- en uitgangssignalen van het applicatieprogramma

De in-/uitgangssignalen van het applicatieprogramma worden onderverdeeld in 3 afzonderlijke databuffers, namelijk:

CIF_GUS[] - uitgangssignalen: gewenste uitgangssturing
(commandering door het applicatieprogramma)

- CIF_WUS[]** - ingangssignalen: werkelijke uitgangssturing
(terugmelding door de procesbesturing)
- CIF_IS[]** - ingangssignalen: detectoren e.d.
(melding door de procesbesturing)

De indeling van de in- en uitgangssignalen dient per regeltoestel te worden gespecificeerd.
De indeling van de CIF_WUS[]-buffer dient gelijk te zijn aan de indeling van de CIF_GUS[]-buffer.

5.2.1 Gewenste uitgangssturing

De gewenste toestand van de uitgangssignalen wordt door het applicatieprogramma in de CIF_GUS[]-buffer van de interface geschreven (zie ook paragraaf 4.2).
De CIF_GUS[]-buffer wordt gedefinieerd als een array van variabelen van het type s_int16. De grootte van de CIF_GUS[]-buffer wordt bepaald door de sommatie van de constanten CIF_AANT_US_FC (aantal fasecycli) en CIF_AANT_US_OV (aantal overige uitgangssignalen). De waarde van CIF_AANT_US_FC en CIF_AANT_US_OV is vastgelegd in de variabelen CIF_PB_AANT_US_FC en CIF_PB_AANT_US_OV beide van het type s_int16.

De procesbesturing leest de uitgangssignalen uit de CIF_GUS[]-buffer.

Voor de gewenste uitgangssturing van de fasecycli geldt de volgende codering:

<u>decimaal</u>	<u>symbool</u>	<u>betekenis</u>
0	CIF_ROOD	Rood / Geel knippen voetgangers
1	CIF_GROEN	Groen / Wit / Wit knippen
2	CIF_GEEL	Geel / Groen Knippen
3	CIF_KNIPWIT	Wit knippen (gereserveerd)
4	CIF_GEDOOFD	Lantaarns gedoofd (gereserveerd)

Voor de gewenste uitgangssturing van de overige uitgangssignalen is de codering vrij en dient per geval te worden vastgelegd.

5.2.2 Werkelijke uitgangssturing

De werkelijke toestand van de uitgangssignalen wordt door de procesbesturing zo snel mogelijk in de CIF_WUS[]-buffer van de interface geschreven.

De CIF_WUS[]-buffer is voor het applicatieprogramma een terugmelding of de gewenste toestand van het uitgangssignaal daadwerkelijk is gerealiseerd.

De CIF_WUS[]-buffer wordt gedefinieerd als een array van variabelen van het type s_int16. De grootte van de CIF_WUS[]-buffer wordt bepaald door de sommatie van de constanten CIF_AANT_US_FC (aantal fasecycli) en CIF_AANT_US_OV (aantal overige uitgangssignalen). De waarde van CIF_AANT_US_FC en CIF_AANT_US_OV is vastgelegd in de variabelen CIF_PB_AANT_US_FC en CIF_PB_AANT_US_OV beide van het type s_int16 (zie ook bij 5.2.1).

Het applicatieprogramma leest de uitgangssignalen uit de CIF_WUS[]-buffer.

Voor de codering van de werkelijke uitgangssturing van de fasecycli geldt dezelfde codering als voor de gewenste uitgangssturing, namelijk:

<u>decimaal</u>	<u>symbool</u>	<u>betekenis</u>
0	CIF_ROOD	Rood / Geel knippen voetgangers
1	CIF_GROEN	Groen / Wit / Wit knippen
2	CIF_GEEL	Geel / Groen Knippen
3	CIF_KNIPWIT	Wit knippen (gereserveerd)
4	CIF_GEDOOFD	Lantaarns gedoofd (gereserveerd)

Voor de werkelijke uitgangssturing van de overige uitgangssignalen is de codering vrij en dient per geval te worden vastgelegd.

5.2.3 Ingangssignalen

De status van de ingangssignalen wordt door de procesbesturing in de **CIF_IS[]**-buffer van de interface geschreven (zie ook paragraaf 4.2).

De **CIF_IS[]**-buffer wordt gedefinieerd als een array van variabelen van het type `s_int16`. De grootte van de **CIF_IS[]**-buffer wordt bepaald door de sommatie van de constanten `CIF_AANT_IS_D` (aantal detectoren) en `CIF_AANT_IS_OV` (aantal overige ingangssignalen). De waarde van `CIF_AANT_IS_D` en `CIF_AANT_IS_OV` is vastgelegd in de variabelen `CIF_PB_AANT_IS_D` en `CIF_PB_AANT_IS_OV` beide van het type `s_int16`.

Het applicatieprogramma leest de ingangssignalen uit de **CIF_IS[]**-buffer.

Voor de ingangssignalen van de detectie geldt de volgende codering:

<u>binair</u>	<u>symbool</u>	<u>betekenis</u>
xxxx xxxx xxxx xxx1	CIF_DET_BEZET	detectie bezet
xxxx xxxx xxxx xx1x	CIF_DET_STORING	detectie storing
xxxx xxxx xxxx x11x	CIF_DET_BOVENGEDRAG	detectie bovengedrag
xxxx xxxx xxxx 1x1x	CIF_DET_ONDERGEDRAG	detectie ondergedrag

De informatie van snelheidsdetectoren wordt vastgelegd in de overige ingangssignalen van de **CIF_IS[]**-buffer.

Voor de ingangssignalen van snelheidsdetectoren geldt de volgende codering:

<u>binair</u>	<u>mogelijkheden</u>	<u>betekenis</u>
xxxx xxxx 1111 1111		voertuigsnelheid (0-255 km/h)
1xxx xxxx xxxx xxxx		rijrichting: 0=normale richting, 1=tegen richting
xxx1 xxxx xxxx xxxx		status: 0=betrouwbaar, 1=onbetrouwbaar
xxxx x111 xxxx xxxx	000	voertuig: geen voertuigpassage/informatie
xxxx x111 xxxx xxxx	001	voertuig: personenauto
xxxx x111 xxxx xxxx	010	voertuig: vrachtwagen
xxxx x111 xxxx xxxx	011	voertuig: bus
xxxx x111 xxxx xxxx	100	voertuig: personenwagen + aanhanger
xxxx x111 xxxx xxxx	101	voertuig: vrachtwagen + aanhanger
xxxx x111 xxxx xxxx	110	voertuig: niet gebruikt
xxxx x111 xxxx xxxx	111	voertuig: ongeldig voertuig

Voor de niet omschreven ingangssignalen van de overige ingangen is de codering vrij en dient deze per geval te worden vastgelegd.

Ten behoeve van een eventueel toekomstige uitbreiding is de naam `CIF_ISL[]` gereserveerd voor een buffer van het type `s_int32`.

5.3 Programmastatus

De programmastatus van het regeltoestel kan worden onderverdeeld in 2 afzonderlijke databuffers van het type `s_int16`, namelijk:

- CIF_GPS[]** - gewenste programmastatus
(commandering van het applicatieprogramma)
- CIF_WPS[]** - werkelijke programmastatus
(terugmelding van de procesbesturing)

De elementen 0 t/m 4 van deze arrays worden gebruikt of gereserveerd voor coderingen volgens dit rapport. De overige elementen van deze arrays zijn vrij te gebruiken.

5.3.1 Gewenste programmastatus

De gewenste programmastatus wordt door het applicatieprogramma in de **CIF_GPS[]**-buffer van de interface geschreven. De **CIF_GPS[]**-buffer wordt gedefinieerd als een array met variabelen van het type `s_int16`. De grootte van de **CIF_GPS[]**-buffer wordt vastgelegd in de variabele

CIF_PB_AANT_GPS van het type `s_int16`.

De procesbesturing leest de gewenste programmastatus zo snel mogelijk uit de **CIF_GPS[]**-buffer.

De gewenste programmastatus kan worden toegeedeeld aan de **CIF_GPS[]**-buffer volgens de volgende indeling:

- CIF_GPS[0]** - Programmastatus
CIF_GPS[1] - Programma fout
CIF_GPS[2] - Gereserveerd
CIF_GPS[3] - Gereserveerd
CIF_GPS[4] - Gereserveerd

Voor **CIF_GPS[0]** geldt de volgende codering:

<u>decimaal</u>	<u>symbool</u>	<u>betekenis</u>
0	CIF_STAT_ONGEDEF	ongedefinieerd (gereserveerd)
1	CIF_STAT_GEDOOFD	gedoofd (gereserveerd)
2	CIF_STAT_KP	knipperen (gereserveerd)
3	CIF_STAT_GEEL	geel bij inschakelen (gereserveerd)
4	CIF_STAT_AR	alles rood (gereserveerd)
5	CIF_STAT_REG	regelen (gereserveerd)

Voor **CIF_GPS[1]** geldt de volgende codering:

<u>binair</u>	<u>symbool</u>	<u>betekenis</u>
xxxx xxxx xxxx xxx1	CIF_FB_FOUT	fasebewaking (gereserveerd)
xxxx xxxx xxxx xx1x	CIF_AP_FOUT2	procesbesturing wacht te lang met lezen van door het applicatieprogramma geschreven informatie in de interface (gereserveerd)
xxxx xxxx xxxx x1xx	CIF_AP_FOUT3	applicatieprogramma leest een niet gedefinieerde code in de interface (gereserveerd)

5.3.2 Werkelijke programmastatus

De werkelijke programmastatus wordt door de procesbesturing zo snel mogelijk in de **CIF_WPS[]**-buffer van de interface geschreven. De **CIF_WPS[]**-buffer wordt gedefinieerd als een array met variabelen van het type `s_int16`. De grootte van de **CIF_WPS[]**-buffer wordt vastgelegd in de variabele **CIF_PB_AANT_WPS** van het type `s_int16`.

Het applicatieprogramma leest de werkelijke programmastatus uit de **CIF_WPS[]**-buffer.

De werkelijke programmastatus kan worden toegeedeeld aan de **CIF_WPS[]**-buffer volgens de volgende indeling:

CIF_WPS[0] - Programmastatus
CIF_WPS[1] - Programmafout
CIF_WPS[2] - Gereserveerd
CIF_WPS[3] - Gereserveerd
CIF_WPS[4] - Gereserveerd

Voor de codering van de **CIF_WPS[0]**-buffer geldt dezelfde codering als voor de **CIF_GPS[0]**-buffer, namelijk:

<u>decimaal</u>	<u>symbool</u>	<u>betekenis</u>
0	CIF_STAT_ONGEDEF	ongedefinieerd
1	CIF_STAT_GEDOOFD	gedoofd
2	CIF_STAT_KP	knippen
3	CIF_STAT_GEEL	geel bij inschakelen
4	CIF_STAT_AR	alles rood
5	CIF_STAT_REG	regelen

Voor **CIF_WPS[1]** geldt de volgende codering:

<u>binair</u>	<u>symbool</u>	<u>betekenis</u>
xxxx xxxx xxxx xxx1	CIF_PB_FOUT1	werkelijke uitgangssturing te lang ongelijk aan gewenste uitgangssturing (gereserveerd)
xxxx xxxx xxxx xx1x	CIF_PB_FOUT2	applicatieprogramma wacht te lang met lezen van door de procesbesturing geschreven informatie in de interface (gereserveerd)
xxxx xxxx xxxx x1xx	CIF_PB_FOUT3	procesbesturing leest een niet gedefinieerde code in de interface (gereserveerd)
xxxx xxxx xxxx 1xxx	CIF_PB_FOUT4	applicatieprogramma duurt te lang (gereserveerd)

5.4 Berichten met uitzondering van berichten voor monitoring

Het uitwisselen van verschillende soorten informatie (tussen het applicatieprogramma en de procesbesturing) kan geschieden met behulp van berichten (characterstrings).

In de interface zijn hiertoe 2 berichtenbuffers gedefinieerd van het type `s_int8`, namelijk:

CIF_UBER[] - databuffer van uitgaande berichten van het applicatieprogramma. De grootte van de **CIF_UBER[]**-buffer wordt vastgelegd in de variabele **CIF_PB_MAX_UBER** van het type `s_int16`.

CIF_IBER[] - databuffer van inkomende berichten voor het applicatieprogramma. De grootte van de **CIF_IBER[]**-buffer wordt vastgelegd in de variabele **CIF_PB_MAX_IBER** van het type `s_int16`.

5.4.1 Uitgaande berichten van het applicatieprogramma

De uitgaande berichten worden door het applicatieprogramma in de berichtenbuffer **CIF_UBER[]** geschreven en worden door de procesbesturing uit de **CIF_UBER[]**-buffer gelezen.

De uitgaande berichten van het applicatieprogramma zullen in het algemeen gegevens zijn van het applicatieprogramma, zoals b.v. de status van de variabelen bij fasebewaking (dump).

De uitgaande berichtenbuffer **CIF_UBER[]** is een zogenaamd ringtellerbuffer en beschikt daartoe over een tweetal pointers, namelijk:

***CIF_UBER_LEES** startadres voor het lezen van berichten

***CIF_UBER_SCHRIJF** startadres voor het schrijven van berichten

De waarde van ***CIF_UBER_LEES** dient door de procesbesturing te worden aangepast na het uitlezen van een bericht en de waarde van ***CIF_UBER_SCHRIJF** dient door het

applicatieprogramma te worden aangepast na het wegschrijven van het bericht in de **CIF_UBER[]**-buffer.

Door het berekenen van het verschil tussen beide pointers kan worden nagegaan of de vrije ruimte groot genoeg is voor het wegschrijven van een bericht.

Data uit de berichtenbuffer wordt door de procesbesturing zo snel mogelijk onbewerkt, één op één doorgegeven naar de RS232c-poort.

5.4.2 Ingaande berichten voor het applicatieprogramma

De ingaande berichten worden door de procesbesturing in de berichtenbuffer **CIF_IBER[]** geschreven en worden door het applicatieprogramma uit de **CIF_IBER[]**-buffer gelezen.

De ingaande berichten voor het applicatieprogramma zullen in het algemeen opdrachten zijn voor het applicatieprogramma, zoals b.v. het opvragen van een dump bij fasebewaking.

De ingaande berichtenbuffer **CIF_IBER[]** is een zogenaamd ringtellerbuffer en beschikt daartoe over een tweetal pointers, namelijk:

***CIF_IBER_LEES** startadres voor het lezen van berichten

***CIF_IBER_SCHRIJF** startadres voor het schrijven van berichten

De waarde van ***CIF_IBER_LEES** dient door het applicatieprogramma te worden aangepast na het uitlezen van een bericht en de waarde van ***CIF_IBER_SCHRIJF** dient door de procesbesturing te worden aangepast na het wegschrijven van het bericht in de **CIF_IBER[]**-buffer. Door het berekenen van het verschil tussen beide pointers kan worden nagegaan of de vrije ruimte groot genoeg is voor het wegschrijven van een bericht.

Data van de RS232c-poort voor de berichtenbuffer wordt door de procesbesturing zo snel mogelijk onbewerkt, één op één doorgegeven.

5.5 Berichten ten behoeve van monitoring

Het uitwisselen van verschillende soorten informatie t.b.v. verkeersmonitoring (tussen het applicatieprogramma en de procesbesturing) kan geschieden met behulp van berichten (character-strings).

In de interface zijn hiertoe 2 berichtenbuffers gedefinieerd van het type `s_int8`, namelijk:

CIF_MON_UBER[] - databuffer van uitgaande berichten van het applicatieprogramma. De grootte van de **CIF_MON_UBER[]**-buffer wordt vastgelegd in de variabele **CIF_PB_MAX_MON_UBER** van het type `s_int16`.

CIF_MON_IBER[] - databuffer van inkomende berichten voor het applicatieprogramma. De grootte van de **CIF_MON_IBER[]**-buffer wordt vastgelegd in de variabele **CIF_PB_MAX_MON_IBER** van het type `s_int16`.

5.5.1 Uitgaande berichten van het applicatieprogramma t.b.v. verkeersmonitoring

De uitgaande berichten t.b.v. verkeersmonitoring worden door het applicatieprogramma in de berichtenbuffer **CIF_MON_UBER[]** geschreven en worden door de procesbesturing uit de **CIF_MON_UBER[]**-buffer gelezen.

De uitgaande berichten t.b.v. verkeersmonitoring van het applicatieprogramma zullen gegevens zijn, zoals b.v. de status van een aantal in- en uitgangssignalen.

De uitgaande berichtenbuffer **CIF_MON_UBER[]** is een zogenaamd ringtellerbuffer en beschikt daartoe over een tweetal pointers, namelijk:

***CIF_MON_UBER_LEES** startadres voor het lezen van berichten

***CIF_MON_UBER_SCHRIJF** startadres voor het schrijven van berichten

De waarde van ***CIF_MON_UBER_LEES** dient door de procesbesturing te worden aangepast na het uitlezen van een bericht en de waarde van ***CIF_MON_UBER_SCHRIJF** dient door het applicatieprogramma te worden aangepast na het wegschrijven van het bericht in de **CIF_MON_UBER[]**-buffer.

Door het berekenen van het verschil tussen beide pointers kan worden nagegaan of de vrije ruimte groot genoeg is voor het wegschrijven van een bericht.

Data uit de berichtenbuffer wordt door de procesbesturing onbewerkt, één op één doorgegeven.

5.5.2 Ingaande berichten voor het applicatieprogramma t.b.v. verkeersmonitoring

De ingaande berichten t.b.v. verkeersmonitoring worden door de procesbesturing in de berichtenbuffer **CIF_MON_IBER[]** geschreven en worden door het applicatieprogramma uit de **CIF_MON_IBER[]**-buffer gelezen.

De ingaande berichten voor het applicatieprogramma t.b.v. verkeersmonitoring zullen in beginsel niet vaak worden gebruikt maar worden voor de volledigheid wel gedefinieerd.

De ingaande berichtenbuffer **CIF_MON_IBER[]** is een zogenaamd ringtellerbuffer en beschikt daartoe over een tweetal pointers, namelijk:

***CIF_MON_IBER_LEES** startadres voor het lezen van berichten

***CIF_MON_IBER_SCHRIJF** startadres voor het schrijven van berichten

De waarde van ***CIF_MON_IBER_LEES** dient door het applicatieprogramma te worden aangepast na het uitlezen van een bericht en de waarde van ***CIF_MON_IBER_SCHRIJF** dient door de procesbesturing te worden aangepast na het wegschrijven van het bericht in de

CIF_MON_IBER[]-buffer. Door het berekenen van het verschil tussen beide pointers kan worden nagegaan of de vrije ruimte groot genoeg is voor het wegschrijven van een bericht.

Data voor de berichtenbuffer wordt door de procesbesturing onbewerkt, één op één doorgegeven.

5.6 Berichten t.b.v. fileopslag van verkeersgegevens

Het wegschrijven van verschillende soorten informatie (van het applicatieprogramma naar de procesbesturing) t.b.v. fileopslag van verkeersgegevens door de procesbesturing kan geschieden met behulp van berichten (characterstrings).

In de interface is hiertoe een apart berichtenbuffer gedefinieerd van het type `s_int8`, namelijk:

CIF_FILE_UBER[] - databuffer van uitgaande berichten van het applicatieprogramma. De grootte van het **CIF_FILE_UBER[]**-buffer wordt vastgelegd in de variabele **CIF_PB_MAX_FILE_UBER** van het type `s_int16`.

In de interface is een vlag gedefinieerd van het type `s_int16` voor het aangeven van de status van fileopslag.

CIF_FILE_UBER_STATUS - vlag voor het aangeven van de status van fileopslag

In de interface is een pointer gedefinieerd van het type `s_int8` en een vlag van het type `s_int16` voor het aangeven van een filewisselpunt (geschikt beginpunt voor een nieuwe file), namelijk:

***CIF_FILE_UBER_FILEWISSEL** - pointer voor het aangeven van een filewisselpunt door het applicatieprogramma

CIF_PB_FILE_UBER_FILEWISSEL - vlag voor aanvraag van een (extra) filewisselpunt door de procesbesturing

In de interface is een karakterbuffer gedefinieerd van het type `s_int8` voor het aangeven van de gewenste file-extensie, namelijk:

CIF_FILE_UBER_FILEEXT[] - gewenste file-extensie (null-terminated string)

5.6.1 Uitgaande berichten van het applicatieprogramma t.b.v. fileopslag

De uitgaande berichten t.b.v. fileopslag van verkeersgegevens worden door het applicatieprogramma in de berichtenbuffer **CIF_FILE_UBER[]** geschreven en worden door het de procesbesturing uit de **CIF_FILE_UBER[]**-buffer gelezen.

De uitgaande berichten t.b.v. fileopslag van verkeersgegevens van het applicatieprogramma zullen gegevens zijn, zoals b.v. de status van in- en uitgangssignalen.

De uitgaande berichtenbuffer **CIF_FILE_UBER[]** is een zogenaamd ringtellerbuffer en beschikt daartoe over een lees- en schrijfpunter, namelijk:

***CIF_FILE_UBER_LEES** startadres voor het lezen van berichten

***CIF_FILE_UBER_SCHRIJF** startadres voor het schrijven van berichten

De waarde van ***CIF_FILE_UBER_LEES** dient door de procesbesturing te worden aangepast na het uitlezen van een bericht en de waarde van ***CIF_FILE_UBER_SCHRIJF** dient door het applicatieprogramma te worden aangepast na het wegschrijven van het bericht in de **CIF_FILE_UBER[]**-buffer.

Door het berekenen van het verschil tussen beide pointers kan worden nagegaan of de vrije ruimte groot genoeg is voor het wegschrijven van een bericht.

Data uit de berichtenbuffer wordt door de procesbesturing onbewerkt, één op één weggeschreven naar file.

De uitgaande berichtenbuffer **CIF_FILE_UBER[]** beschikt over een vlag voor het aangeven van de status van fileopslag:

CIF_FILE_UBER_STATUS vlag voor het aangeven van de status van fileopslag

De status van de vlag **CIF_FILE_UBER_STATUS** wordt “geset” door het applicatieprogramma (**CIF_FILE_UBER_STATUS= 1**) indien het applicatieprogramma fileopslag door de procesbesturing wenst. De vlag wordt “gereset” door het applicatieprogramma (**CIF_FILE_UBER_STATUS= 0**) indien het applicatieprogramma geen fileopslag wenst.

5.6.2 Filewisselpunt

De uitgaande berichtenbuffer **CIF_FILE_UBER[]** beschikt over een pointer en een vlag voor het aangeven en aanvragen van een filewisselpunt (geschikt beginpunt van een nieuwe file), namelijk:

***CIF_FILE_UBER_FILEWISSEL** pointer voor het aangeven van een filewisselpunt door het applicatieprogramma

CIF_PB_FILE_UBER_FILEWISSEL vlag voor aanvraag van een (extra) filewisselpunt door de procesbesturing

De waarde van ***CIF_FILE_UBER_FILEWISSEL** dient door het applicatieprogramma te worden aangepast voor het aangeven van een filewisselpunt. Het applicatieprogramma geeft hiermee een punt binnen de uitgaande berichtenbuffer **CIF_FILE_UBER[]** aan waar het beginpunt is voor een logische filewissel. Na het lezen van het karakter van het filewisselpunt dient de waarde van

***CIF_FILE_UBER_FILEWISSEL** te worden gereset” door de procesbesturing (**CIF_FILE_UBER_FILEWISSEL= 0**). Indien **CIF_FILE_UBER_FILEWISSEL** de waarde nul heeft is er geen filewisselpunt aanwezig.

Het applicatieprogramma bepaalt tijdens verkeerslogging periodiek (bijvoorbeeld iedere 5 minuten) een filewisselpunt. De procesbesturing kan met de vlag **CIF_PB_FILE_UBER_FILEWISSEL** een aanvraag doen aan het applicatieprogramma voor het creëren van een extra filewisselpunt.

Een aanvraag voor een filewisselpunt wordt “geset” door de procesbesturing (**CIF_PB_FILE_UBER_FILEWISSEL= 1**). Na het schrijven van een filewisselpunt wordt de vlag “gereset” door het applicatieprogramma (**CIF_PB_FILE_UBER_FILEWISSEL= 0**).

5.6.3 File-extensie

De file-extensie kan worden vastgelegd in de karakterbuffer **CIF_FILE_UBER_FILEEXT[]**. **CIF_FILE_UBER_FILEEXT[]** is een zogenaamde “null terminated string”; dit betekent dat de “string” moet worden afgesloten door de waarde 0 (‘\0’-karakter). Bij initialisatie van het regelprogramma wordt de waarde van **CIF_FILE_UBER_FILEEXT[]** door het applicatieprogramma bepaald. De file-extensie wordt gebruikt door de procesbesturing bij de opslag van de files in het verkeersregeltoestel. De file-extensie mag 1, 2, of 3 karakters groot zijn (exclusief het ‘\0’-karakter).

5.7 Systeemklok

De interface bevat een systeemklok ten behoeve van het applicatieprogramma. Deze systeemklok wordt door de procesbesturing bijgehouden en voor elke aanroep van de functie *applicatieprogramma()* in de **CIF_KLOK[]**-buffer geschreven.

Het applicatieprogramma leest de systeemklok uit de **CIF_KLOK[]**-buffer.

De systeemklok bevat de datum, de dag van de week, de tijd en een aantal systeemtijdtellers (secondenteller, tiende secondenteller).

De klokbuffer **CIF_KLOK[]** wordt gedefinieerd als een array met variabelen van het type `s_int16`. De grootte van de **CIF_KLOK[]**-buffer wordt vastgelegd in de variabele **CIF_PB_AANT_KLOK** van het type `s_int16`. De **CIF_KLOK[]**-buffer is als volgt opgebouwd:

CIF_KLOK[0] - jaar
CIF_KLOK[1] - maand
CIF_KLOK[2] - dag
CIF_KLOK[3] - dagsoort (zo t/m za)
CIF_KLOK[4] - uur
CIF_KLOK[5] - minuut
CIF_KLOK[6] - seconde
CIF_KLOK[7] - secondenteller
CIF_KLOK[8] - tienden secondenteller

De systeemtijdtellers die in de **CIF_KLOK[]**-buffer zijn opgenomen kunnen worden gebruikt voor de opbouw van de tijdelementen in het applicatieprogramma.

Voor **CIF_KLOK[3]** gelden de volgende waarden:

<u>decimaal</u>	<u>symbool</u>
0	CIF_ZONDAG
1	CIF_MAANDAG
2	CIF_DINSDAG
3	CIF_WOENSDAG
4	CIF_DONDERDAG
5	CIF_VRIJDAG
6	CIF_ZATERDAG

Voor **CIF_KLOK[7]** en **CIF_KLOK[8]** geldt dat deze niet synchroon behoeven te lopen met **CIF_KLOK[0]** t/m **CIF_KLOK[6]**; tevens geldt voor **CIF_KLOK[7]** en **CIF_KLOK[8]** dat na het bereiken van de waarde 32767 weer vanaf 0 wordt geteld.

5.8 Parameters

Het applicatieprogramma maakt gebruik van zogenaamde parametervariabelen voor het aangeven van de waarden van tijden, tellers, etc. De waarden van deze parametervariabelen worden in de interface opgenomen.

De interface onderscheidt 2 typen parameterbuffers, namelijk:

CIF_PARM1[] - parameters van het type `s_int16`

CIF_PARM2[] - parameters van het type `s_int32`

De grootte van de **CIF_PARM1[]**-buffer wordt vastgelegd in de variabele **CIF_PB_AANT_PARM1** van het type `s_int16`.

De grootte van de **CIF_PARM2[]**-buffer wordt vastgelegd in de variabele **CIF_PB_AANT_PARM2** van het type `s_int16`.

Afhankelijk van de gewenste maximum waarde voor een parametervariabele zal gebruik worden gemaakt van **CIF_PARM1[]** of **CIF_PARM2[]**.

Wijziging van de parameters kan op 2 manieren geschieden:

1. via het applicatieprogramma
2. via de procesbesturing

Uitgangspunt bij beide methoden is dat de parameterinstellingen beheerd worden binnen de procesbesturing.

Per buffer wordt een variabele opgenomen die kan aangeven welk array-element is gewijzigd, dit om het bijwerken van gewijzigde parameters niet onnodig lang te laten duren.

Het applicatieprogramma of de procesbesturing "reset" na het lezen van de gewijzigde parameter de betreffende variabele.

Indien op initiatief van het applicatieprogramma een parameterinstelling wordt gewijzigd, schrijft het applicatieprogramma de nieuwe instelling op de desbetreffende plaats in de interface.

De procesbesturing bevestigt het lezen daarvan.

Indien op initiatief van de procesbesturing een parameterinstelling wordt gewijzigd, schrijft de procesbesturing de nieuwe instelling op de desbetreffende plaats in de interface.

Het applicatieprogramma bevestigt het lezen daarvan.

Indien in de procesbesturing, separaat van de interface, de parameterinstellingen zijn opgeslagen, dan dienen vanuit de procesbesturing die maatregelen te worden getroffen die garanderen dat beide instellingen met elkaar in overeenstemming zijn.

De indeling van de parameterbuffers kan per regeltoestel worden vastgelegd.

Opmerking: Indien gewenst kan het aantal typen worden uitgebreid bijvoorbeeld:

CIF_PARM3[] - parameters van het type `float`

5.9 Selectieve detectie

Na ontvangst van een correct selectief detectiebericht wordt de selectieve detectie informatie door de procesbesturing zo snel mogelijk in de **CIF_DSI[]**-buffer van de interface geschreven. De **CIF_DSI[]**-

buffer wordt gedefinieerd als een array van variabelen van het type s_int16. De grootte van de CIF_DSI[]-buffer wordt vastgelegd in de variabele CIF_PB_AANT_DSI van het type s_int16.

Het applicatieprogramma leest de selectieve detectie informatie uit de CIF_DSI[]-buffer.

De selectieve detectie informatie wordt door de procesbesturing in de CIF_DSI[]-buffer geschreven volgens de indeling in tabel 1 (Selectieve voertuiginformatie). In de tabel is per variabele tevens het mogelijke bereik en de coderingen aangegeven. De selectieve detectie informatie is altijd één systeemronde beschikbaar voor het applicatieprogramma. Indien een variabele niet wordt gebruikt dan wordt door de procesbesturing de gedefinieerde waarde voor geen informatie op de interface gezet (indien bijvoorbeeld geen stiptheidsinformatie beschikbaar is, wordt de waarde 3600 ingevuld).

Per systeemronde kan de informatie van één selectief voertuig van de procesbesturing worden doorgegeven aan het applicatieprogramma. De procesbesturing zorgt zonnodig voor buffering om ervoor te zorgen dat er geen voertuiginformatie verloren gaat.

Indien geen nieuwe selectieve detectie informatie beschikbaar is, worden door de procesbesturing de gedefinieerde waarden voor geen informatie in de CIF_DSI[]-buffer geschreven.

Tabel 1 - Selectieve voertuig informatie

Nr	Variabele	Bereik	Codering
0	lus/meldpunt(volg)nummer	0 - 127	0 - geen informatie
1	Voertuigcategorie	0 - 99	0 - geen informatie 1 - bus 2 - tram 3 - politie 4 - brandweer 5 - ambulance 6 - CVV 7 - taxi 8-70 gereserveerd 71-98 vrij te gebruiken 99 - URO (onbekend voertuig)
2	lijnnummer openbaar vervoer	0 - 9999	0 - geen informatie
3	wagendienstnummer	0 - 9999	0 - geen informatie
4	(uniek) bedrijfsnummer	0 - 255	0 - geen informatie
5	(uniek) voertuignummer (per bedrijf) (groot wagennummer bij OV)	0 - 32767	0 - geen informatie
6	richtingsaanduiding, ook volgrichtingen	0 - 255	0 - geen informatie 1 t/m 200 fasecyclusnummer 201 - rechtsaf 202 - linksaf 203 - rechtdoor 204-255 - gereserveerd
7	Voertuigstatus	0 - 99	0 - geen informatie 1 - rijden 2 - halteren 3 - vertrekmelding 4 - stilstand 5-99 - gereserveerd
8	Prioriteitsklasse	0 - 99	0 - geen informatie 1 - geen prioriteit (b.v. alleen aanvraag) 2 - geconditioneerd 3 - absolute prioriteit 4 - zwaailicht/sirene 5-99 - gereserveerd
9	Stiptheidsklasse	0 - 99	0 - geen informatie 1 - te laat 2 - op tijd 3 - te vroeg 4 - niet volgens dienstregeling 5-99 - gereserveerd
10	stiptheid [s]	-3600 tot 3600	3600 - geen informatie <0 - te vroeg >0 - te laat
11	voertuiglengte [m]	0 - 255	0 - geen informatie

12	voertuigsnelheid [m/s]	0 - 99	99 - geen informatie
13	afstand tot stopstreep [m]	-99 tot 9999	9999 - geen informatie <0 - na de stopstreep >0 - voor de stopstreep
14	passeertijd stopstreep [s]	0 - 255	255 - geen informatie
15	ritnummer	0 - 9999	0 - geen informatie
16	ritcategorie	0 - 99	0 - geen informatie 1-9 - versterkingsrit 10 - dienstregelingsrit 11 - materieelrit 12 - inrukrit (naar remise) 13 - uitrukrit (vanaf remise) 14-99 - gereserveerd
17	Route openbaar vervoer	0 - 99	0 - geen informatie 1 - route 1 (route a) 2 - route 2 (route b) 3-99 - vrij te gebruiken
18	type melding	0 - 99	0 - geen informatie 1 - inmelding 2 - uitmelding 3 - voormelding 4-99 - gereserveerd
19	codering meldpunt(volg)nummer (OV-bedrijf)	0 - 32767	0 - geen informatie
20	locatie-aanduiding Breedtegraad graden	0 - 89	Locatie conform WGS84
21	locatie-aanduiding Breedtegraad minuten	0 - 59	Locatie conform WGS84
22	locatie-aanduiding Breedtegraad seconden	0 - 59	Locatie conform WGS84
23	locatie-aanduiding Breedtegraad honderdste sec.	0 - 99	Locatie conform WGS84
24	locatie-aanduiding Lengtegraad graden	0 - 179	Locatie conform WGS84
25	locatie-aanduiding Lengtegraad minuten	0 - 59	Locatie conform WGS84
26	locatie-aanduiding Lengtegraad seconden	0 - 59	Locatie conform WGS84
27	locatie-aanduiding Lengtegraad honderdste sec.	0 - 99	Locatie conform WGS84
28	jaartal	0 - 9999	actuele datum/tijd verzending bericht
29	maand	1 - 12	
30	dag	1 - 31	
31	uren	0 - 23	
32	minuten	0 - 59	
33	seconden	0 - 59	
34	reserve	0 - 32767	0 - geen informatie
35	reserve	0 - 32767	0 - geen informatie

5.10 Lezen en schrijven in de interface

Voor het lezen en schrijven van de in de interface gedefinieerde buffers worden vlaggen opgenomen, die aangeven dat de inhoud van een buffer gewijzigd is. Deze vlaggen zijn variabelen van het type s_int16.

Voor de waarden van deze vlaggen geldt de volgende codering:

<u>decimaal</u>	<u>symbool</u>	<u>betekenis</u>
0	CIF_GELEZEN	buffer gelezen
1	CIF_GESCHREVEN	in buffer geschreven

De volgende vlaggen worden onderscheiden:

CIF_ISWIJZ WIJZIGEN INGANGSSIGNALLEN

- wordt "geset" door de procesbesturing na het schrijven van de ingangssignalen (CIF_ISWIJZ=1)
- wordt "gereset" door het applicatieprogramma na het lezen van de ingangssignalen (CIF_ISWIJZ=0)

Zolang CIF_ISWIJZ nog niet is "gereset" geldt voor variabelen voor detectie in de IS-buffer dat als een bit in de IS-buffer de waarde '1' heeft dit bit niet in de waarde '0' mag worden gewijzigd.

CIF_WUSWIJZ WIJZIGEN WERKELIJKE UITGANGSSTURING

1. wordt "geset" door de procesbesturing na het schrijven van de werkelijke uitgangssturing (CIF_WUSWIJZ=1)
2. wordt "gereset" door het applicatieprogramma na het lezen van de werkelijke uitgangssturing (CIF_WUSWIJZ=0)

CIF_WPSWIJZ WIJZIGEN WERKELIJKE STATUS

1. wordt "geset" door de procesbesturing na het schrijven van de werkelijke status (CIF_WPSWIJZ=1)
2. wordt "gereset" door het applicatieprogramma na het lezen van de werkelijke status (CIF_WPSWIJZ=0)

CIF_GUSWIJZ WIJZIGEN GEWENSTE UITGANGSSTURING

1. wordt "geset" door het applicatieprogramma na het schrijven van de gewenste uitgangssturing (CIF_GUSWIJZ=1)
2. wordt "gereset" door de procesbesturing na het lezen van de gewenste uitgangssturing (CIF_GUSWIJZ=0)

CIF_GPSWIJZ WIJZIGEN GEWENSTE STATUS

1. wordt "geset" door het applicatieprogramma na het schrijven van de gewenste status (CIF_GPSWIJZ=1)
2. wordt "gereset" door de procesbesturing na het lezen van de gewenste status (CIF_GPSWIJZ=0)

CIF_DSIWIJZ WIJZIGEN SELECTIEVE DETECTIE INFORMATIE

1. wordt "geset" door de procesbesturing na het schrijven van de selectieve detectie informatie (CIF_DSIWIJZ=1) indien er een nieuw selectief voertuig is gedetecteerd
2. wordt "gereset" door het applicatieprogramma na het lezen van de selectieve detectie informatie (CIF_DSIWIJZ=0)

De bovenstaande werkwijze houdt in dat soms informatie verloren gaat.

Voor de normale toepassing van signaalgroepen en detectoren zal dit geen problemen geven (processnelheid).

Indien dit niet gewenst is zullen afwijkende oplossingen gekozen moeten worden.

Voor het lezen en schrijven van de in de interface gedefinieerde parameters worden er per buffer twee vlaggen opgenomen met de volgende functies:

1. een vlag die aangeeft dat de inhoud van een parameter gewijzigd is door de procesbesturing.
2. een vlag die aangeeft dat de inhoud van een parameter gewijzigd is door het applicatieprogramma.

In deze twee vlaggen kan het arrayelementnummer van de gewijzigde parameter worden geschreven, dit om bijwerken van de parameters niet onnodig lang te laten duren.

De vlaggen zijn variabelen van het type `s_int16`.

Voor de waarden van deze vlaggen geldt de volgende codering:

<u>decimaal</u>	<u>symbool</u>	<u>betekenis</u>
-1	CIF_GEEN_PARMWIJZ	geen wijzigingen in parameterbuffer
-2	CIF_MEER_PARMWIJZ	meerdere wijzigingen in parameterbuffer
-3	CIF_INIT_PARM	initiële waarden in parameterbuffer

De volgende vlaggen worden onderscheiden:

CIF_PARM1WIJZPB WIJZIGEN PARAMETERS DOOR PROCESBESTURING

1. wordt "geset" door de procesbesturing na het schrijven van een gewijzigde parameter in CIF_PARM1[]. (CIF_PARM1WIJZPB=arraynummer). Indien meerder parameters zijn gewijzigd krijgt CIF_PARM1WIJZPB de waarde -2.
2. wordt "gereset" door het applicatieprogramma na lezen van de gewijzigde parameter(s). (CIF_PARM1WIJZPB = -1).

CIF_PARM1WIJZAP WIJZIGEN PARAMETERS DOOR APPLICATIEPROGRAMMA

1. wordt "geset" door het applicatieprogramma na het schrijven van een gewijzigde parameter in CIF_PARM1[]. (CIF_PARM1WIJZAP=arraynummer). Indien meerdere parameters zijn gewijzigd krijgt CIF_PARM1WIJZAP de waarde -2. Indien alle initiële waarden van parameters in de CIF_PARM1-buffer zijn geschreven krijgt CIF_PARM1WIJZAP de waarde -3.
2. wordt "gereset" door de procesbesturing na lezen van de gewijzigde parameter(s). (CIF_PARM1WIJZAP = -1).

CIF_PARM2WIJZPB WIJZIGEN PARAMETERS DOOR PROCESBESTURING

1. wordt "geset" door de procesbesturing na het schrijven van een gewijzigde parameter in CIF_PARM2[] (CIF_PARM2WIJZPB=arraynummer). Indien meerder parameters zijn gewijzigd krijgt CIF_PARM2WIJZPB de waarde -2.
2. wordt "gereset" door het applicatieprogramma na lezen van de gewijzigde parameter(s). (CIF_PARM2WIJZPB = -1).

CIF_PARM2WIJZAP WIJZIGEN PARAMETERS DOOR APPLICATIEPROGRAMMA

1. wordt "geset" door het applicatieprogramma na het schrijven van een gewijzigde parameter in CIF_PARM2[]. (CIF_PARM2WIJZAP=arraynummer). Indien meerdere parameters zijn gewijzigd krijgt CIF_PARM2WIJZAP de waarde -2. Indien alle initiële waarden van parameters in de CIF_PARM1-buffer zijn geschreven krijgt CIF_PARM2WIJZAP de waarde -3.
2. wordt "gereset" door de procesbesturing na lezen van de gewijzigde parameter(s). (CIF_PARM2WIJZAP = -1).

5.11 Ontruimingstijden

Volgens paragraaf 4.3 dient de procesbesturing naast de garantie-ontruimingstijden te werken met ontruimingstijden.

De waarden van de ontruimingstijden kunnen op twee manieren aan de procesbesturing bekend worden gemaakt:

1. via de interface en
2. separaat van de interface

Ad1.

Indien de waarden van de ontruimingstijden via de interface aan de procesbesturing kenbaar worden gemaakt dient:

- de indeling van de interface aan de kant van de procesbesturing bekend te zijn en
- wordt door middel van de variabele CIF_OT van het type s_int16 aan de procesbesturing bekend gemaakt dat de betreffende waarden de enige maatgevende waarden zijn voor de binnen de procesbesturing te hanteren ontruimingstijden (CIF_OT = 1).

Ad2.

Indien de waarden van de ontruimingstijden separaat van de interface aan de procesbesturing kenbaar worden gemaakt dient:

- middels een apart document een lijst met waarden van ontruimingstijden beschikbaar te zijn en
- wordt door middel van de onder Ad 1 genoemde variabele CIF_OT aan de procesbesturing bekend gemaakt dat de betreffende waarden de enige maatgevende waarden zijn voor de binnen de procesbesturing te hanteren ontruimingstijden en dat geen gebruik mag worden gemaakt van de eventueel in de interface aanwezige waarden voor ontruimingstijden (CIF_OT = 0).

6 SLOTOPMERKINGEN

6.1 Realisatie van de C-interface

1. Bij een realisatie van de C-interface in een verkeersregeltoestel dient alles wat in document is beschreven te worden geïmplementeerd.
2. Naast het in dit document genoemde kunnen in de pakketten CCOL en RWS C-regelaar pakketafhankelijke buffers als uitbreiding op de C-interface worden gedefinieerd. Deze pakketafhankelijke buffers zullen worden aangeduid met namen die van de volgende voorvoegsels zijn voorzien:
CIF_CCOL_ voor het CCOL pakket en
CIF_RWSC_ voor het pakket RWS C-regelaar.
De gebruiker zal zelf dienen te regelen in hoeverre deze pakketafhankelijke buffers bij een realisatie van de C-interface in een verkeersregeltoestel moeten worden meegenomen. Een voorbeeld van een mogelijke toepassing is het weergeven van de interne signaalgroepstoestand in de interface.
Pakketafhankelijke buffers zullen uitsluitend door de pakketeigenaars voor hun eigen pakket worden vastgesteld en doorgevoerd.

6.2 Reactietijd

Het is gewenst dat een gegeven applicatieprogramma tijdens de toestand regelen tenminste 1x per 0,1 seconde wordt doorlopen.

Bovenstaande geldt alleen indien

- geen berichtenverkeer plaats vindt en
- er niet wordt geïnitieerd.

6.3 Nader te regelen

Als gevolg van hetgeen in dit document is beschreven dient de gebruiker de onderstaande zaken nader te regelen:

- Het wel of niet aanwezig zijn van de RS232c-poort in het regeltoestel.
- De instellingen voor de communicatie via de RS232c-poort.
- De kruispunafhankelijke gegevens zoals bedoelt paragraaf 4.1.
- De indeling van de CIF_GUS[]-buffer en de CIF_IS[]-buffer.
- De codering van de overige in- en uitgangssignalen (indien gebruikt).
- De aanvullende codering voor de ingangssignalen van de detectie (indien gebruikt).
- De codering van de overige elementen van de CIF_GPS[]-buffer en de CIF_WPS[]-buffer (indien gebruikt).
- De gewenste uitwerking van de gebruikte coderingen die in dit document als gereserveerd zijn aangegeven.
- De indeling van de parameterbuffers voor zover noodzakelijk.
- Het indien gewenst laten verzorgen van functies voor tellingen, dichtheidsmetingen, snelheidsmetingen etc.

7 VERKLARING GEBRUIKTE SYMBOLEN

CIF_GUS[]	-	gewenste uitgangssturing (commandering door applicatieprogramma)
CIF_WUS[]	-	werkelijke uitgangssturing (terugmelding van de procesbesturing)
CIF_PB_AANT_US_FC	-	aantal fasecycli in CIF_GUS-buffer en CIF_WUS-buffer
CIF_PB_AANT_US_OV	-	aantal overige uitgangen in CIF_GUS en CIF_WUS-buffer
CIF_IS[]	-	ingangssignalen: detectoren e.d. (melding van de procesbesturing)
CIF_PB_AANT_IS_D	-	aantal detectoren in CIF_IS-buffer
CIF_PB_AANT_IS_OV	-	aantal overige ingangen in CIF_IS-buffer
CIF_DSI[]	-	selectieve detectie informatie
CIF_PB_AANT_DSI	-	grootte CIF_DSI-buffer
CIF_GPS[]	-	gewenste programmastatus (commandering door het applicatieprogramma)
CIF_PB_AANT_GPS	-	grootte CIF_GPS-buffer
CIF_WPS[]	-	werkelijke programmastatus (terugmelding van de procesbesturing)
CIF_PB_AANT_WPS	-	grootte CIF_WPS-buffer
CIF_IBER[]	-	ingående berichten voor het applicatieprogramma
CIF_PB_MAX_IBER	-	grootte CIF_IBER-buffer
*CIF_IBER_LEES	-	pointer voor het lezen van berichten
*CIF_IBER_SCHRIJF	-	pointer voor het schrijven van berichten
CIF_UBER[]	-	uitgaande berichten van het applicatieprogramma
CIF_PB_MAX_UBER	-	grootte CIF_UBER-buffer
*CIF_UBER_LEES	-	pointer voor het lezen van berichten
*CIF_UBER_SCHRIJF	-	pointer voor het schrijven van berichten
CIF_MON_IBER[]	-	ingående berichten voor het applicatieprogramma t.b.v. verkeers- monitoring
CIF_PB_MAX_MON_IBER	-	grootte CIF_MON_IBER-buffer
*CIF_MON_IBER_LEES	-	pointer voor het lezen van berichten
*CIF_MON_IBER_SCHRIJF	-	pointer voor het schrijven van berichten
CIF_MON_UBER[]	-	uitgaande berichten van het applicatieprogramma t.b.v. verkeersmonitoring
CIF_PB_MAX_MON_UBER	-	grootte CIF_MON_UBER-buffer
*CIF_MON_UBER_LEES	-	pointer voor het lezen van berichten
*CIF_MON_UBER_SCHRIJF	-	pointer voor het schrijven van berichten
CIF_FILE_UBER[]	-	uitgaande berichten van het applicatieprogramma t.b.v. fileopslag van verkeersgegevens
CIF_PB_MAX_FILE_UBER	-	grootte CIF_FILE_UBER-buffer
*CIF_FILE_UBER_LEES	-	pointer voor het lezen van berichten
*CIF_FILE_UBER_SCHRIJF	-	pointer voor het schrijven van berichten
CIF_FILE_UBER_STATUS	-	status van fileopslag

*CIF_FILE_UBER_FILEWISSEL - pointer voor het aangeven van een filewisselpunt
 CIF_PB_FILE_UBER_FILEWISSEL - aanvraag van procesbesturing voor een (extra) filewisselpunt

CIF_FILE_UBER_FILEEXT[] - gewenste file-extensie (null-terminated string)

CIF_KLOK[] - systeemklok voor het applicatieprogramma
 CIF_PB_AANT_KLOK - grootte CIF_KLOK-buffer

CIF_PARM1[] - parametervariabelen type s_int16
 CIF_PB_AANT_PARM1 - grootte CIF_PARM1-buffer

CIF_PARM2[] - parametervariabelen type s_int32
 CIF_PB_AANT_PARM2 - grootte CIF_PARM2-buffer

CIF_ISWIJZ - ingangssignalen gewijzigd
 CIF_WUSWIJZ - werkelijke uitgangssturing gewijzigd
 CIF_WPSWIJZ - werkelijke programmastatus gewijzigd
 CIF_GUSWIJZ - gewenste uitgangssturing gewijzigd
 CIF_GPSWIJZ - gewenste programmastatus gewijzigd
 CIF_PARM1WIJZPB - parameters door procesbesturing gewijzigd
 CIF_PARM1WIJZAP - parameters door applicatieprogramma gewijzigd
 CIF_PARM2WIJZPB - parameters door procesbesturing gewijzigd
 CIF_PARM2WIJZAP - parameters door applicatieprogramma gewijzigd

CIF_OT - gebruik ontruimingstijden in interface

8 IMPLEMENTATIE

Voor implementatie van de interface in het regeltoestel dient gebruik gemaakt te worden van het met de applicatie programmatuur mee te leveren bestand "CIF.INC".

In dit bestand worden de voor de interface benodigde variabelen, definities en functie gedefinieerd of gedeclareerd.

De in het bestand "CIF.INC" gebruikte macrodefinitie CIF_PUBLIC en de constanten CIF_AANT_US_FC, CIF_AANT_US_OV, CIF_AANT_IS_D, CIF_AANT_IS_OV, CIF_AANT_GPS, CIF_AANT_WPS, CIF_MAX_UBER, CIF_MAX_IBER, CIF_AANT_KLOK, CIF_AANT_PARM1 en CIF_AANT_PARM2 en indien van toepassing de constanten CIF_AANT_DSI, CIF_MAX_MON_IBER, CIF_MAX_MON_UBER, CIF_MAX_FILE_UBER en CIF_MAX_FILEEXT dienen in het applicatieprogramma te worden gedefinieerd.

Indien ontruimingstijden via de interface aan de procesbesturing kenbaar gemaakt worden dient de in het bestand "CIF.INC" daartoe te gebruikten macrodefinitie CIF_WEL_OT in het applicatieprogramma te worden gedefinieerd.

Onderstaand is de inhoud van dit bestand weergegeven.

```
/*
 * CVN Commissie C
 * bestand: CIF.INC
 * versie 12 oktober 2011
 */

typedef char   s_int8;      /* signed char 8 bits          */
typedef short  s_int16;     /* signed integer 16 bits     */
typedef long   s_int32;     /* signed integer 32 bits     */

#ifdef CIF_PUBLIC

    s_int16 CIF_GUS [CIF_AANT_US_FC + CIF_AANT_US_OV]; /* CIF_GUS[ ]-buffer          */
    const s_int16 CIF_PB_AANT_US_FC = CIF_AANT_US_FC; /* aantal fasecycli in CIF_GUS- en
    CIF_WUS-buffer          */
    const s_int16 CIF_PB_AANT_US_OV = CIF_AANT_US_OV; /* aantal overige uitgangen in
    CIF_GUS- en CIF_WUS-buffer          */

    s_int16 CIF_WUS [CIF_AANT_US_FC + CIF_AANT_US_OV]; /* CIF_WUS[ ]-buffer          */

    s_int16 CIF_IS [CIF_AANT_IS_D + CIF_AANT_IS_OV]; /* CIF_IS[ ]-buffer          */
    const s_int16 CIF_PB_AANT_IS_D = CIF_AANT_IS_D; /* aantal detectoren in CIF_IS
    buffer */
    const s_int16 CIF_PB_AANT_IS_OV = CIF_AANT_IS_OV; /* aantal overige ingangen in CIF_IS
    buffer          */

    s_int16 CIF_DSI[CIF_AANT_DSI]; /* CIF_DSI[] buffer          */
    const s_int16 CIF_PB_AANT_DSI= CIF_AANT_DSI; /* grootte CIF_DSI buffer    */
    s_int16 CIF_DSIWIJZ ; /* vlag CIF_DSI[]-buffer gewijzigd */

    s_int16 CIF_GPS [CIF_AANT_GPS]; /* CIF_GPS[ ]-buffer          */
    const s_int16 CIF_PB_AANT_GPS = CIF_AANT_GPS; /* grootte CIF_GPS buffer    */

    s_int16 CIF_WPS [CIF_AANT_WPS]; /* CIF_WPS[ ]-buffer          */
    const s_int16 CIF_PB_AANT_WPS = CIF_AANT_WPS; /* grootte CIF_WPS buffer    */

    s_int8 CIF_UBER [CIF_MAX_UBER]; /* CIF_UBER[ ]-buffer          */
    const s_int16 CIF_PB_MAX_UBER= CIF_MAX_UBER; /* grootte CIF_UBER buffer    */

    s_int8 CIF_IBER [CIF_MAX_IBER]; /* CIF_IBER[ ]-buffer          */
    const s_int16 CIF_PB_MAX_IBER= CIF_MAX_IBER; /* grootte CIF_IBER buffer    */

    s_int8 *CIF_UBER_LEES; /* pointer naar te lezen bericht */
#endif
```

```

s_int8 *CIF_UBER_SCHRIJF; /* pointer naar te schrijven bericht */

s_int8 *CIF_IBER_LEES; /* pointer naar te lezen bericht */
s_int8 *CIF_IBER_SCHRIJF; /* pointer naar te schrijven bericht */

s_int8 CIF_MON_IBER [CIF_MAX_MON_IBER]; /* CIF_MON_IBER[]-buffer */
const s_int16 CIF_PB_MAX_MON_IBER= CIF_MAX_MON_IBER; /* grootte CIF_MON_IBER buffer */

s_int8 *CIF_MON_IBER_LEES; /* pointer naar te lezen bericht */
s_int8 *CIF_MON_IBER_SCHRIJF; /* pointer naar te schrijven bericht */

s_int8 CIF_MON_UBER [CIF_MAX_MON_UBER]; /* CIF_MON_UBER[]-buffer */
const s_int16 CIF_PB_MAX_MON_UBER= CIF_MAX_MON_UBER; /* grootte CIF_MON_UBER buffer */

s_int8 *CIF_MON_UBER_LEES; /* pointer naar te lezen bericht */
s_int8 *CIF_MON_UBER_SCHRIJF; /* pointer naar te schrijven bericht */

s_int8 CIF_FILE_UBER[CIF_MAX_FILE_UBER]; /* CIF_FILE_UBER[ ] - buffer */
const s_int16 CIF_PB_MAX_FILE_UBER= CIF_MAX_FILE_UBER; /* grootte CIF_FILE_UBER buffer */

s_int8 *CIF_FILE_UBER_LEES; /* pointer naar te lezen bericht */
s_int8 *CIF_FILE_UBER_SCHRIJF; /* pointer naar te schrijven bericht */

s_int16 CIF_FILE_UBER_STATUS; /*status van fileopslag */

s_int8 *CIF_FILE_UBER_FILEWISSEL; /* pointer voor het aangeven van een filewisselpunt */
s_int16 CIF_PB_FILE_UBER_FILEWISSEL; /* aanvraag voor een (extra) filewisselpunt */

s_int8 CIF_FILE_UBER_FILEEXT[CIF_MAX_FILEEXT]; /* gewenste file-extensie */
/* (null-terminated string) */

s_int16 CIF_KLOK [CIF_AANT_KLOK]; /* CIF_KLOK[ ]-buffer */
const s_int16 CIF_PB_AANT_KLOK= CIF_AANT_KLOK; /* grootte CIF_KLOK buffer */

s_int16 CIF_PARM1 [CIF_AANT_PARM1]; /* CIF_PARM1[ ]-buffer */
const s_int16 CIF_PB_AANT_PARM1 = CIF_AANT_PARM1; /* grootte CIF_PARM1 buffer */

s_int32 CIF_PARM2 [CIF_AANT_PARM2]; /* CIF_PARM2[ ]-buffer */
const s_int16 CIF_PB_AANT_PARM2 = CIF_AANT_PARM2; /* grootte CIF_PARM2 buffer */

s_int16 CIF_ISWIJZ; /* vlag CIF_IS[ ]-buffer gewijzigd */
s_int16 CIF_WUSWIJZ; /* vlag CIF_WUS[ ]-buffer gewijzigd */
s_int16 CIF_WPSWIJZ; /* vlag CIF_WPS[ ]-buffer gewijzigd */
s_int16 CIF_GUSWIJZ; /* vlag CIF_GUS[ ]-buffer gewijzigd */
s_int16 CIF_GPSWIJZ; /* vlag CIF_GPS[ ]-buffer gewijzigd */
s_int16 CIF_PARM1WIJZPB; /* vlag CIF_PARM1[ ]-buffer gewijzigd door */
s_int16 CIF_PARM2WIJZPB; /* vlag CIF_PARM2[ ]-buffer gewijzigd door */
s_int16 CIF_PARM1WIJZAP; /* vlag CIF_PARM1[ ]-buffer gewijzigd door */
s_int16 CIF_PARM2WIJZAP; /* vlag CIF_PARM2[ ]-buffer gewijzigd door */

#ifdef CIF_WEL_OT
const s_int16 CIF_OT = 1; /* Gebruik ontruimingstijden in de interface */
#else
const s_int16 CIF_OT = 0; /* Geen gebruik ontruimingstijden in de interface */
#endif

#else

extern s_int16 CIF_GUS[ ]; /* CIF_GUS[ ]-buffer */
extern const s_int16 CIF_PB_AANT_US_FC; /* aantal fasecycli in CIF_GUS- en CIF_WUS-buffer */
extern const s_int16 CIF_PB_AANT_US_OV; /* aantal overige uitgangen in CIF_GUS- en CIF_WUS-buffer */

extern s_int16 CIF_WUS [ ]; /* CIF_WUS[ ]-buffer */

extern s_int16 CIF_IS [ ]; /* CIF_IS[ ]-buffer */
extern const s_int16 CIF_PB_AANT_IS_D; /* aantal detectoren in CIF_IS buffer */
extern const s_int16 CIF_PB_AANT_IS_OV; /* aantal overige ingangen in CIF_IS buffer */

```



```

extern s_int16 CIF_DSI[]; /* CIF_DSI[] buffer */
extern const s_int16 CIF_PB_AANT_DSI; /* grootte CIF_DSI buffer */
extern s_int16 CIF_DSIWIJZ ; /* vlag CIF_DSI[]-buffer gewijzigd */

extern s_int16 CIF_GPS [ ]; /* CIF_GPS[ ]-buffer */
extern const s_int16 CIF_PB_AANT_GPS; /* grootte CIF_GPS buffer */

extern s_int16 CIF_WPS [ ]; /* CIF_WPS[ ]-buffer */
extern const s_int16 CIF_PB_AANT_WPS; /* grootte CIF_WPS buffer */

extern s_int8 CIF_UBER [ ]; /* CIF_UBER[ ]-buffer */
extern const s_int16 CIF_PB_MAX_UBER; /* grootte CIF_UBER buffer */

extern s_int8 CIF_IBER [ ]; /* CIF_IBER[ ]-buffer */
extern const s_int16 CIF_PB_MAX_IBER; /* grootte CIF_IBER buffer */

extern s_int8 *CIF_UBER_LEES; /* pointer naar te lezen bericht */
extern s_int8 *CIF_UBER_SCHRIJF; /* pointer naar te schrijven bericht */

extern s_int8 *CIF_IBER_LEES; /* pointer naar te lezen bericht */
extern s_int8 *CIF_IBER_SCHRIJF; /* pointer naar te schrijven bericht */

extern s_int8 CIF_MON_IBER [ ]; /* CIF_MON_IBER[]-buffer */
extern const s_int16 CIF_PB_MAX_MON_IBER; /* grootte CIF_MON_IBER buffer */

extern s_int8 *CIF_MON_IBER_LEES; /* pointer naar te lezen bericht */
extern s_int8 *CIF_MON_IBER_SCHRIJF; /* pointer naar te schrijven bericht */

extern s_int8 CIF_MON_UBER [ ]; /* CIF_MON_UBER[]-buffer */
extern const s_int16 CIF_PB_MAX_MON_UBER; /* grootte CIF_MON_UBER buffer */

extern s_int8 *CIF_MON_UBER_LEES; /* pointer naar te lezen bericht */
extern s_int8 *CIF_MON_UBER_SCHRIJF; /* pointer naar te schrijven bericht */

extern s_int8 CIF_FILE_UBER [ ]; /* CIF_FILE_UBER[]-buffer */
extern const s_int16 CIF_PB_MAX_FILE_UBER; /* grootte CIF_FILE_UBER buffer */

extern s_int8 *CIF_FILE_UBER_LEES; /* pointer naar te lezen bericht */
extern s_int8 *CIF_FILE_UBER_SCHRIJF; /* pointer naar te schrijven bericht */

extern s_int16 CIF_FILE_UBER_STATUS; /*status van fileopslag */

extern s_int8 *CIF_FILE_UBER_FILEWISSEL; /* pointer voor het aangeven van een filewisselpunt */
extern s_int16 CIF_PB_FILE_UBER_FILEWISSEL; /* aanvraag voor een(extra) filewisselpunt */

extern s_int8 CIF_FILE_UBER_FILEEXT[]; /* gewenste file-extensie */
/* (null-terminated string) */

extern s_int16 CIF_KLOK [ ]; /* CIF_KLOK[ ]-buffer */
extern const s_int16 CIF_PB_AANT_KLOK; /* grootte CIF_KLOK buffer */

extern s_int16 CIF_PARM1 [ ]; /* CIF_PARM1[ ]-buffer */
extern const s_int16 CIF_PB_AANT_PARM1; /* grootte CIF_PARM1 buffer */

extern s_int32 CIF_PARM2 [ ]; /* CIF_PARM2[ ]-buffer */
extern const s_int16 CIF_PB_AANT_PARM2; /* grootte CIF_PARM2 buffer */

extern s_int16 CIF_ISWIJZ; /* vlag CIF_IS[ ]-buffer gewijzigd */
extern s_int16 CIF_WUSWIJZ; /* vlag CIF_WUS[ ]-buffer gewijzigd */
extern s_int16 CIF_WPSWIJZ; /* vlag CIF_WPS[ ]-buffer gewijzigd */
extern s_int16 CIF_GUSWIJZ; /* vlag CIF_GUS[ ]-buffer gewijzigd */
extern s_int16 CIF_GPSWIJZ; /* vlag CIF_GPS[ ]-buffer gewijzigd */
extern s_int16 CIF_PARM1WIJZPB; /* vlag CIF_PARM1[ ]-buffer gewijzigd door */
extern s_int16 CIF_PARM2WIJZPB; /* vlag CIF_PARM2[ ]-buffer gewijzigd door */
extern s_int16 CIF_PARM1WIJZAP; /* vlag CIF_PARM1[ ]-buffer gewijzigd door */
extern s_int16 CIF_PARM2WIJZAP; /* vlag CIF_PARM2[ ]-buffer gewijzigd door */

extern const s_int16 CIF_OT; /* Gebruik ontruimingstijden in interface */

#endif

```

```

/* CIF_GUS[ ] en CIF_WUS[ ] */
#define CIF_ROOD 0 /* rood */
#define CIF_GROEN 1 /* groen / wit / wit knippen */
#define CIF_GEEL 2 /* geel / groen knippen */
#define CIF_KNIPWIT 3 /* wit knippen */
#define CIF_GEDOOFD 4 /* lantaarn(s) gedoofd */

/* CIF_IS[ ] */
#define CIF_DET_BEZET 1 /* detectie bezet */
#define CIF_DET_STORING 2 /* detectie storing */
#define CIF_DET_BOVENGEDRAG 4 /* detectie bovengedrag */
#define CIF_DET_ONDERGEDRAG 8 /* detectie ondergedrag */

/* CIF_GPS [ ] en CIF_WPS [ ] */
#define CIF_PROG_STATUS 0 /* CIF_GPS en CIF_WPS */
/* programmastatus buffer */
#define CIF_PROG_FOUT 1 /* CIF_GPS en CIF_WPS */
/* programma-foutbuffer */

/* CIF_GPS [CIF_PROG_STATUS] */
/* en CIF_WPS [CIF_PROG_STATUS] */
#define CIF_STAT_ONGEDEF 0 /* ongedefinieerd */
#define CIF_STAT_GEDOOFD 1 /* gedoofd */
#define CIF_STAT_KP 2 /* knippen */
#define CIF_STAT_GEEL 3 /* geel bij inschakelen */
#define CIF_STAT_AR 4 /* alles rood */
#define CIF_STAT_REG 5 /* regelen */

/* CIF_GPS [CIF_PROG_FOUT] */
#define CIF_GEEN_FOUT 0 /* geldt ook voor */
/* CIF_WPS [CIF_PROG_FOUT] en */
/* return-waarde */
/* applicatieprogramma() */
#define CIF_FB_FOUT 1 /* fasebewaking */
#define CIF_AP_FOUT2 2 /* procesbesturing wacht te lang met */
/* lezen van door het applicatie- */
/* programma geschreven informatie */
#define CIF_AP_FOUT3 4 /* applicatieprogramma leest een */
/* niet gedefinieerde code */

/* CIF_WPS [CIF_PROG_FOUT] */
#define CIF_PB_FOUT1 1 /* werkelijke uitgangssturing te lang */
/* ongelijk aan gewenste uitgangs- */
/* sturing */
#define CIF_PB_FOUT2 2 /* applicatieprogramma wacht te lang */
/* met lezen van door de proces- */
/* besturing geschreven informatie */
#define CIF_PB_FOUT3 4 /* procesbesturing leest een niet */
/* gedefinieerde code */
#define CIF_PB_FOUT4 8 /* applicatieprogramma duurt te lang */

/* CIF_KLOK[ ] */
#define CIF_JAAR 0 /* jaartal */
#define CIF_MAAND 1 /* maand 1..12 */
#define CIF_DAG 2 /* dag van de maand 1..31 */
#define CIF_DAGSOORT 3 /* dag van de week 0..6 */
#define CIF_UUR 4 /* uur van de dag 0..23 */
#define CIF_MINUUT 5 /* minuut 0..59 */
#define CIF_SECONDE 6 /* seconde 0..59 */
#define CIF_SEC_TELLER 7 /* seconden teller 0..32767 */
#define CIF_TSEC_TELLER 8 /* tienden sec teller 0..32767 */

/* CIF_KLOK[DAGSOORT] */
#define CIF_ZONDAG 0 /* zondag */
#define CIF_MAANDAG 1 /* maandag */
#define CIF_DINSDAG 2 /* dinsdag */
#define CIF_WOENSDAG 3 /* woensdag */
#define CIF_DONDERDAG 4 /* donderdag */
#define CIF_VRIJDAG 5 /* vrijdag */
#define CIF_ZATERDAG 6 /* zaterdag */

```

```

/* lees en schrijf vlaggen */
#define CIF_GESCHREVEN 1 /* in buffer geschreven */
#define CIF_GELEZEN 0 /* buffer gelezen */
#define CIF_GEEN_PARMWIJZ -1 /* geen wijzigingen in /*
/* parameterbuffer */
#define CIF_MEER_PARMWIJZ -2 /* meerdere wijzigingen in /*
/* parameterbuffer */
#define CIF_INIT_PARM -3 /* initiele waarden in /*
/* parameterbuffer */

/* aanroep en returnwaarde applicatieprogramma() */
#define CIF_GEEN_INIT 0 /* 'normale' aanroep */
#define CIF_INIT 1 /* init na programmastart */
/*
#define CIF_GEEN_FOUT 0 zie CIF_GPS[CIF_PROG_FOUT
*/
#define CIF_INIT_FOUT -1 /* initialisatie fout */

/* CIF_DSI */
/* codering van de elementen */
/* ----- */
#define CIF_DSI_LUS 0 /* lusnummer */
#define CIF_DSI_VTG 1 /* voertuigcategorie */
#define CIF_DSI_LYN 2 /* lijnnummer */
#define CIF_DSI_WDNST 3 /* wagen dienstnummer */
#define CIF_DSI_BEDR 4 /* bedrijfsnummer */
#define CIF_DSI_NUM 5 /* voertuignummer */
#define CIF_DSI_DIR 6 /* richtingsnummer */
#define CIF_DSI_STAT 7 /* voertuigstatus */
#define CIF_DSI_PRI 8 /* prioriteitsklasse */
#define CIF_DSI_STP 9 /* stiptheidsklasse */
#define CIF_DSI_TSTP 10 /* stiptheid [s] */
#define CIF_DSI_LEN 11 /* voertuiglengte [m] */
#define CIF_DSI_SPD 12 /* voertuigsnelheid [m/s] */
#define CIF_DSI_LSS 13 /* afstand tot stopstreep [m] */
#define CIF_DSI_TSS 14 /* passeertijd stopstreep [s] */
#define CIF_DSI_RIT 15 /* ritnummer */
#define CIF_DSI_RITC 16 /* ritcategorie */
#define CIF_DSI_ROUT 17 /* routenummer openbaar vervoer */
#define CIF_DSI_TYPE 18 /* type melding */
#define CIF_DSI_OVC 19 /* codering meldpunt
/* openbaar vervoer
#define CIF_DSI_XGRAD 20 /* breedtegraad graden
#define CIF_DSI_XMIN 21 /* breedtegraad minuten
#define CIF_DSI_XSEC 22 /* breedtegraad seconden
#define CIF_DSI_XHSEC 23 /* breedtegraad honderdste seconden
#define CIF_DSI_YGRAD 24 /* lengtegraad graden
#define CIF_DSI_YMIN 25 /* lengtegraad minuten
#define CIF_DSI_YSEC 26 /* lengtegraad seconden
#define CIF_DSI_YHSEC 27 /* lengtegraad honderdste seconden
#define CIF_DSI_JAAR 28 /* jaartal
#define CIF_DSI_MND 29 /* maand
#define CIF_DSI_DAG 30 /* dag
#define CIF_DSI_UUR 31 /* uur
#define CIF_DSI_MIN 32 /* minuten
#define CIF_DSI_SEC 33 /* seconden
#define CIF_DSI_RES1 34 /* reserve 1
#define CIF_DSI_RES2 35 /* reserve 2

/* afwijkende defaultwaarden */
/* ----- */
#define CIF_DSI_TSTP_DEF 3600 /* defaultwaarde CIF_DSI_TSTP */
#define CIF_DSI_LSS_DEF 9999 /* defaultwaarde CIF_DSI_LSS */
#define CIF_DSI_TSS_DEF 255 /* defaultwaarde CIF_DSI_TSS */

/* voertuigcategorie */
/* ----- */
#define CIF_BUS 1 /* bus */
#define CIF_TRAM 2 /* tram */
#define CIF_POL 3 /* politie */
#define CIF_BRA 4 /* brandweer */
#define CIF_AMB 5 /* ambulance */
#define CIF_CVV 6 /* collectief vraagafhankelijk*/

```

```

/* vervoer */
#define CIF_TAXI      7      /* taxi */
#define CIF_URO      99     /* onbekend voertuig */

/* richting */
/* ----- */
#define CIF_RAF      201    /* rechtsaf */
#define CIF_LAF      202    /* linksaf */
#define CIF_RDR      203    /* rechtdoor */

/* voertuigstatus */
/* ----- */
#define CIF_RIJD      1      /* rijden */
#define CIF_HALT      2      /* halteren */
#define CIF_START     3      /* vertrekmelding */
#define CIF_STOP      4      /* stilstand */

/* prioriteit */
/* ----- */
#define CIF_PRI1      1      /* geen prioriteit */
/* alleen aanvraag */
#define CIF_PRI2      2      /* geconditioneerde prioriteit*/
#define CIF_PRI3      3      /* absolute prioriteit */
#define CIF_SIR      4      /* sirene/zwaailicht */

/* stiptheid */
/* ----- */
#define CIF_TE_LAAT   1      /* te laat */
#define CIF_OP_TIJD   2      /* op tijd */
#define CIF_TE_VROEG  3      /* te vroeg */

/* ritcategorie */
/* ----- */
#define CIF_DIENST    10     /* dienstregelingsrit */
#define CIF_MAT       11     /* materieelrit */
#define CIF_MATIN     12     /* materieelrit inrukrit */
/* naar remise */
#define CIF_MATUIT    13     /* materieelrit uitrukrit */
/* van remise */

/* type melding */
/* ----- */
#define CIF_DSIN      1      /* inmelding */
#define CIF_DSUIT     2      /* uitmelding */
#define CIF_DSVOR     3      /* voormelding */

/* declaratie functie applicatieprogramma */
s_int16 applicatieprogramma (s_int16);

```