

Instituut voor Cultuurtechniek en Waterhuishouding
Wageningen

ASPECTEN VAN INFORMATIEVERWERKING

26

VERWERKING VAN AUTOMATISCHE GEREgistREERDE GEGEVENS

MET HET PROGRAMMAPAKKET LOGGER

K. Oostindie

**BIBLIOTHEEK
STARINGGEBOUW**

Nota's van het Instituut zijn in principe interne communicatie-
middelen, dus geen officiële publikaties.

Hun inhoud varieert sterk en kan zowel betrekking hebben op een
eenvoudige weergave van cijferreeksen, als op een concluderende
discussie van onderzoeksresultaten. In de meeste gevallen zullen
de conclusies echter van voorlopige aard zijn omdat het onderzoek
nog niet is afgesloten.

Bepaalde nota's komen niet voor verspreiding buiten het Instituut
in aanmerking

ISBN 165239 - 02



ASPECTEN VAN INFORMATIEVERWERKING

26

De nota's handelende over Aspecten van Informatieverwerking bevatten inlichtingen over de ontwikkeling van de informatieverwerking binnen het Instituut. Naast meer concluderende en toelichtende beschouwingen zal aandacht worden besteed aan het gebruik van programma's en programmapakketten en zullen zakelijke inlichtingen over praktijkervaring met en toepassing van de informatieverwerking worden gegeven

I N H O U D

	Blz.
1. INLEIDING	1
2. DATALOGGERS	2
3. HET REGISTRATIEFORMAT VAN DE DATALOGGER	2
4. VERBETEREN VAN DE REGISTRATIES MET BEHULP VAN EEN EDITOR	4
5. VERWERKING VAN DE REGISTRATIES	9
5.1. Het splitsen van de registratiefile	9
5.2. Het verbeteren van eenvoudige registratiefouten	12
5.3. Het wisselen van kanalen	14
5.4. Het testen van de metingen op intervallen	15
5.5. Het transformeren en tabelleren van de meetwaarden	17
6. HET AANKOPPELEN VAN EEN NIEUWE TRANSFORMATIE EN TABELLERINGSSUBROUTINE	18
7. OVERGANG NAAR HET PAKKET 'VAZAL'	21
8. BIJLAGEN	21
REFERENTIES	22
BIJLAGEN	

1. INLEIDING

In deze nota in de reeks Aspecten van Informatieverwerking wordt het programma 'LOGGER', voor de verwerking van automatisch geregistreerde gegevens toegelicht. De ontwikkeling van dit programma vond plaats in opdracht van de Stuurgroep Programma-ontwikkeling Automatische Registratieverwerking (SPAR).

De ideeën over de vorm van het programma zijn ontwikkeld door: ir. G.J.A. Nieuwenhuis, K. Oostindie en ir. J.G. Wesseling, leden van de SPAR.

De automatische registraties die met het programma moeten worden verwerkt zijn afkomstig van dataloggers. Deze registraties werden in een vroeger stadium verwerkt met het programmapakket 'Numerieke Exploratie'. Gezien de grote hoeveelheden gegevens en het toenemen van het aantal dataloggers, werd binnen de stuurgroep de wens geuit, om een programma te ontwikkelen, specifiek gericht op de verwerking van de gegevens afkomstig van zelfregistrerende apparatuur. Het programma is ontwikkeld en operationeel op de PDP 11/70 computer van het TWR in Utrecht.

De samenstelling van de Stuurgroep Programma ontwikkeling Automatische Registratieverwerking was:

dr. Ph.Th. Stol (voorzitter)
ing. J.B.H.M. van Gils (secretaris)
ir. D. Boels
ir. G.J.A. Nieuwenhuis
K. Oostindie
ir. J.G. Wesseling

2. DATALOGGERS

Een datalogger is een zelfregistrerend apparaat. Indien regelmatig elektrische signalen moeten worden vastgelegd afkomstig van verschillende instrumenten wordt dikwijls gebruik gemaakt van een datalogger. Een dergelijk instrument 'scant' op vastgestelde tijden de meetwaarden van de betreffende instrumenten. Deze worden tijdelijk opgeslagen in een buffer, waarna registratie op een of ander randapparaat plaats vindt. Dit laatste is dikwijls een cassetteband. Naast de registraties van metingen wordt meestal gelijktijdig dagaanduiding, tijd en proefnummer geregistreerd.

De op de cassetteband geregistreerde gegevens kunnen weer met behulp van de daartoe geschikte randapparatuur van een computer worden uitgelezen, waarna de informatie op een (magnetische) schijf van het computersysteem wordt vastgelegd. Hiervandaan kan men de registraties in leesbare vorm op papier afdrukken. Dit noemt men dan wel de 'ruwe data' van een datalogger. Deze data kunnen dan met behulp van het programma 'LOGGER' worden verwerkt. Hierbij worden zowel controles uitgevoerd op de metingen als bewerkingen waardoor de gewenste output wordt verkregen. Daar het registreren van de datalogger geheel automatisch verloopt, behoeft een onderzoeker slechts periodiek te controleren of het apparaat goed registreert. Bovendien dient op tijd de cassetteband te worden verwisseld.

3. HET REGISTRATIEFORMAT VAN DE DATALOGGER

De inputgegevens voor het programma LOGGER, moet aan bepaalde eisen voldoen, zodat er enige aandacht besteed moet worden aan de manier van registreren. De gegevens bestaan achtereenvolgens uit: een proefnummer, een dagnummer, het uur en de minuten. Vervolgens n-maal een kanaalnummer met een meetwaarde. Wanneer een datalogger op een bepaald tijdstip de kanalen met hun meetwaarden heeft geregistreerd, noemen we dit een 'scan'. Een scan kan uit een zeker aantal records of regels bestaan. De inhoud van een cassetteband noemen we een registratiefile. Een registratiefile bestaat dus uit x-maal een

scan. Door de Stuurgroep Programma-ontwikkeling Automatische Registratieverwerking zijn met de Technische en Fysische Dienst voor de Landbouw afspraken gemaakt over het registratieformat van de datalogger. Deze afspraken staan vermeld in bijlage 2. Om met het programma-pakket te kunnen werken moet aan een aantal voorwaarden omtrent het registreren worden voldaan. Deze voorwaarden zijn:

1. het eerste record moet achtereenvolgens bevatten: proefnummer, dagnummer, uur, minuut;
2. tweede en eventueel volgende records moeten achtereenvolgens bevatten: n-maal een kanaalnummer met een meetwaarde;
3. alle waarden moeten onderling gescheiden zijn door 1 of meer blanks of komma's;
4. alle waarden moeten uit numerieke characters bestaan;
5. elk record moet worden afgesloten met een slash(/);
6. wanneer een scan bestaat uit n-records dan moet het tweede tot en met de (n-1)ste record uit eenzelfde aantal kanaalnummer met meetwaarden bestaan. Het laatste record van een scan mag minder kanaalnummers met meetwaarden bevatten;
7. Per scan moeten de kanalen met de bijbehorende meetwaarden in dezelfde volgorde worden geregistreerd.

Voorbeeld 1 is ontleend aan de datalogger 'BORSSELE'. De gegevens bestaan uit een gedeelte van een registratiefile. Dit gedeelte van de registratiefile bestaat uit drie scan's en iedere scan bestaat uit vijf records of regels. In dit voorbeeld is in de eerste scan het proefnummer 04214, het dagnummer 042, het uur 15 en de minuten 00. Daarna volgt 23-maal een kanaalnummer met een meetwaarde. Bovendien is elke scan nog afgesloten door een extra slash. Deze extra afsluiting van de scan is geen noodzakelijkheid.

Voorbeeld 1. Registraties afkomstig van een datalogger

```
04214,042 15 00/
020 +0123,021 +0388,022 +0399,023 +0402,024 +0253,025 *****,030 *****/
031 *****,032 *****,033 *****,034 *****,035 *****,001 +0319,002 +0680/
003 +0749,007 +0717,008 +0257,010 +0596,011 +0741,012 +0605,014 +0539/
015 +0618,016 +0572//
04214,042 16 00/
020 +0123,021 +0389,022 +0400,023 +0403,024 +0255,025 *****,030 +0126/
031 +0332,032 +0310,033 +0210,034 *****,035 *****,001 +0319,002 +0681/
003 +0749,007 +0705,008 +0256,010 +0597,011 +0742,012 +0607,014 +0541/
015 +0618,016 +0579//
04214,042 16 59/
020 +0123,021 +0390,022 +0401,023 +0402,024 +0255,025 *****,030 *****/
031 *****,032 *****,033 *****,034 *****,035 *****,001 +0318,002 +0681/
003 +0750,007 +0705,008 +0256,010 +0599,011 +0742,012 +0608,014 +0543/
015 +0620,016 +0580//
```

4. VERBETEREN VAN DE REGISTRATIES MET BEHULP VAN EEN EDITOR

Zoals uit voorbeeld 1 blijkt, is er een zekere regelmaat in recordlengte te zien. Wanneer de gehele registratie file op papier wordt uitgeprint, kan men visueel controleren, of deze regelmaat van recordlengte zich over de gehele file voortzet. Door technische storingen, zoals stroomuitval, kan deze regelmaat van recordlengte onderbroken zijn. Daar het programma 'LOGGER' niet opgezet is om dergelijke grove fouten in de registraties te herstellen, zal dan eerst deze registratiefile met behulp van een editor moeten worden verbeterd. Zoals in voorbeeld 1 te zien is, zijn er een aantal meetwaarden onbekend. Deze meetwaarden worden aangeduid met: *****. Dit zijn alphanumerieke characters welke omgezet worden in een onbekend code, welke door het programma kan worden verwerkt. De onbekend code voor het programma 'LOGGER' is het kleinste integer getal dat op de gebruikte computer kan worden geformeerd, namelijk -32767.

Het komt ook voor, zoals in voorbeeld 2 het geval is, dat een logger zo geconstrueerd is dat aan het eind van elk record geen slash wordt weggeschreven, waarbij opgemerkt wordt dat er wel een 'return'-teken aanwezig is voor overgang naar het volgende record. Dit teken wordt bij het zichtbaar maken van de gegevens niet afgebeeld. De slashes moeten nu met behulp van de editor worden aangebracht. Elk

returnteken moet nu worden vervangen door een slash met een returnteken.

Voorbeeld 2. Registraties van een datalogger, waarbij aan het einde van elk record geen slash voorkomt

```
022 270 10 00
01 +0128 02 +0129 03 +0131 04 +0131 05 +0134 06 +0132
07 +0133 08 +0136 09 +0134 10 +0136 11 +0136 12 +0136
13 +0136 14 +0137 15 -1236 16 +0141 17 +0138 18 +0145
19 +0156 20 +0149 21 +0001 22 +0388 23 +0064 24 +0163
25 +0028 26 +0199 27 +0733 28 +0169 29 +0066 30 +0062
31 +0005 32 +0010 33 +0120 34 +0417 35 +0776
022 270 10 30
01 +0128 02 +0129 03 +0131 04 +0131 05 +0134 06 +0132
07 +0133 08 +0136 09 +0134 10 +0136 11 +0137 12 +0138
13 +0136 14 +0138 15 -1236 16 +0143 17 +0140 18 +0147
19 +0158 20 +0151 21 +0000 22 +0319 23 +0045 24 +0115
25 +0020 26 +0201 27 +0733 28 +0120 29 +0067 30 +0065
31 +0006 32 +0012 33 +0039 34 +0369 35 +0777
```

Voorbeeld 2 is ontleend aan de registraties van de datalogger 'REMOTE SENSING'. Het is een gedeelte van een registratiefile en bestaat uit twee scan's. De twee scan's bestaan ieder weer uit zeven records. In de eerste scan is het proefnummer 022, het dagnummer 270, het uur 10 en de minuten zijn 00. Vervolgens zijn er 35 kanalen met hun meetwaarden geregistreerd.

TECO is een geschikte editor om een returnteken te vervangen door een slash met een returnteken. Dit is een character georiënteerde editor. Enkele voorname commando's van deze editor zullen worden besproken.

Met <RET> wordt het gebruik van de returntoets aangeduid.

Het aanroepen van de editor TECO:

TEC<RET>

De editor TECO wordt opgeroepen en komt na elk ingetikt commando terug met een *. Elk commando moet worden afgesloten door tweemaal achtereen de escapetoets in te drukken. De returntoets kan voor het beëindigen van een commando niet worden gebruikt, want een 'return' wordt beschouwd als een character. Wanneer de escapetoets wordt inge-

drukt, zal op het beeldscherm een dollarteken (\$) worden afgebeeld. Dit teken wordt ook wel alt-mode genoemd. Een opdracht mag op een nieuwe regel worden vervolgd (zie voorbeeld 3). TECO deelt de file, die in de editor wordt geplaatst, in pagina's in. Er wordt door TECO dan een pagina ingelezen en plaatst deze pagina in het kerngeheugen in zijn editing buffer. Nadat de nodige wijzigingen zijn aangebracht, wordt deze pagina naar de TECO-outputfile geschreven en kan de volgende pagina van de TECO-inputfile in de 'buffer' worden gezet.

De registratiefile moet nu in de editfile worden gezet en ook moet er een nieuwe outputfile worden gespecificeerd. Dit kan als volgt:

```
ERlogger.reg$EWnieuw.reg$$
```

```
ER          = Edit Read
```

```
Logger.reg = registratiefile
```

```
EW          = Edit Write
```

```
nieuw.reg  = nieuwe.output registratiefile
```

Dit zijn eigenlijk twee aparte commando's die gescheiden zijn door een alt-mode. Dit noemt men ook wel een command string. Er mogen altijd verschillende commando's achter elkaar worden gegeven, mits deze commando's worden gescheiden door een alt-mode.

De editor TECO werkt met een pointer, welke bijhoudt waar men zich in de registratiefile bevindt. Wanneer men die pointer ook op het beeldscherm wil zien, moet men de volgende opdracht geven:

```
2ET94EV$$
```

Nu zal na elk uitgevoerd commando de regel worden afgebeeld, waarin de pointer zich bevindt. De pointer zelf wordt afgebeeld door: ^ De pointer staat altijd tussen twee characters in. Verder kent TECO onder andere de volgende commando's:

<u>Opdracht</u>	<u>Betekenis</u>
P	Er wordt een nieuwe pagina in de buffer gezet
J	Pointer gaat naar het begin van de buffer
ZJ	Pointer gaat naar het einde van de buffer
L	Pointer gaat naar het begin van de volgende regel
nL	Pointer gaat n regels naar beneden
-nL	Pointer gaat n regels naar boven
Stekst	Pointer zoekt vanaf de plaats, waar hij in de edit file staat het stukje tekst op en plaatst de pointer na dit stukje tekst. De tekst kan ook betrekking hebben op numerieke characters
T	De regel wordt getypt vanaf de pointer tot het einde van de regel
nT	De tekst wordt vanaf de pointer getypt tot de n-de regel na de pointer
HT	De tekst in de buffer wordt in zijn geheel getypt
Itekst	Deze opdracht voert de characters 'tekst' in na de pointer. 'tekst' kan ook betrekking hebben op numerieke characters
nD	Deze opdracht verwijdert n characters na de pointer
-nD	Deze opdracht verwijdert n characters voor de pointer
K	Deze opdracht verwijdert de rest van de regel na de pointer
nK	Deze opdracht verwijdert alles vanaf de pointer tot aan het n-de carriage-return-line-feed teken
HK	Deze opdracht verwijdert alle tekst in de buffer
FStekst1\$tekst2	Zoek het stukje tekst: 'tekst1' op en vervang dit door 'tekst2'
nFStekst1\$tekst2	Zoek de n-de voorkomst van stukje tekst: 'tekst1' op en vervang dit door 'tekst2'
<FStekst1\$tekst2\$>;	Zoek alle stukjes tekst: tekst1 op en vervang dit door tekst2
EX	schrijft de tekst in de buffer naar de outputfile en verlaat de editor TECO

De hierboven beschreven commando's worden het meest gebruikt. Zoals in voorbeeld 2 te zien is ontbreken de slashes aan het einde van ieder record. In voorbeeld 3 wordt aangegeven hoe deze slashes in de registratiefile worden aangebracht.

Voorbeeld 3. Het vervangen van een returnteken door een 'slash-return' met behulp van de editor TECO

```
>TEC<RET>
*ERVVOORB2.DEN$EWVVOORB2.DEN$$
*2ET94EV$$
^*P$$
^022 270 20 00
*<FS<RET>
$/<RET>
$>$$$

%Search fail in iter

%SN1      ; not in iteration
^022 270 10 00/
*HT$$
022 270 10 00/
01 +0128 02 +0129 03 +0131 04 +0131 05 +0134 06 +0132/
07 +0133 08 +0136 09 +0134 10 +0136 11 +0136 12 +0136/
13 +0136 14 +0137 15 -1236 16 +0141 17 +0138 18 +0145/
19 +0156 20 +0149 21 +0001 22 +0388 23 +0064 24 +0163/
25 +0028 26 +0199 27 +0733 28 +0169 29 +0066 30 +0062/
31 +0005 32 +0010 33 +0120 34 +0417 35 +0776/
022 270 10 30/
01 +0128 02 +0129 03 +0131 04 +0131 05 +0134 06 +0132/
07 +0133 08 +0136 09 +0134 10 +0136 11 +0137 12 +0138/
13 +0136 14 +0138 15 -1236 16 +0143 17 +0140 18 +0147/
19 +0158 20 +0151 21 +0000 22 +0319 23 +0045 24 +0115/
25 +0020 26 +0201 27 +0733 28 +0120 29 +0067 30 +0065/
31 +0006 32 +0012 33 +0039 34 +0369 35 +0777/
^022 270 10 00/
*EX$$
```

De inputfile is de registratiefile VOORB2.DEN, als outputfile wordt dezelfde file gebruikt. Dit betekent niet dat de inputfile na het editten verloren is, maar de outputfile krijgt een hogere versienummer mee en de inputfile blijft bewaard. De opdrachten die achter-eenvolgens worden opgegeven zijn:

- Het oproepen van de editor
- Het definiëren van de inputfile en de outputfile
- Het zichtbaar maken van de pointer op het beeldscherm
- Het inzetten van een nieuwe pagina in de buffer
- Het vervangen van returntekens door slashes met returntekens.

Na het laatste returnteken dat de editor gevonden heeft, blijft de editor doorzoeken naar nog een returnteken. Daar TECO nu geen returntekens meer vinden kan wordt hierover een foutmelding gegeven

- Het uittypen van de gehele tekst die zich in de buffer bevindt
- Het verlaten van de editor

5. VERWERKING VAN DE REGISTRATIES

Nadat de registraties met de editor verbeterd zijn, kan begonnen worden met de verdere bewerking. Dit gebeurt met het programma `LOGGER`. Dit programma werkt conversationeel. Conversationeel wil zeggen: het programma stelt vragen en de gebruiker typt de antwoorden in, in het algemeen nadat op het scherm een '?' is verschenen. Wanneer het programma gestart wordt, zal het programma vragen wat u doen wilt. Het programma is opgebouwd uit een hoofdprogramma met een aantal subroutines. Er zijn dan 5 mogelijkheden, te weten:

1. Het splitsen van de registratiefile
2. Het testen van de metingen op intervallen
3. Het verwisselen van kanalen
4. Het transformeren en daarna tabelleren van de meetwaarden
5. Stoppen

Voordat men de mogelijkheden 2, 3 of 4 wil gebruiken, moet eerst mogelijkheid 1 gebruikt zijn. Na elk onderdeel kan het programma worden gestopt om dan op een later tijdstip te worden vervolgd.

5.1. Het splitsen van de registratiefile

Wanneer men alleen de registratiefile heeft, zullen eerst met behulp van het programma gesplitste files gemaakt moeten worden, de zogenaamde splitsfiles. Daarna kunnen verdere bewerkingen op de gege-

vens worden toegepast. Een splitsfile wordt dus uit een registratiefile gemaakt. Een splitsfile bevat per record nooit meer dan 80 characters. Op deze manier kunnen de splitsfiles met behulp van de terminal worden bekeken, zonder dat de lengte van de records storend werkt op het visueel beoordelen van de getallen op het beeldscherm. Het eerste record van elke splitsfile bevat de kanaalnummers waarvan de metingen, behorende bij die kanaalnummers, in de volgende records voorkomen. De eerste splitsfile bevat in het eerste record, buiten de al genoemde kanaalnummers, ook nog een testgetal en het totaal aantal geregistreerde kanalen. In het tweede en de volgende records komen dus de meetwaarden voor. In de eerste splitsfile worden deze meetwaarden vooraf gegaan door het proefnummer, de dag, het uur en de minuten. De eerste splitsfile bevat ook nog de meetwaarden van zes kanalen. De volgende splitsfiles bevatten de meetwaarden van tien kanalen. Nu kan men dus zelf berekenen hoeveel splitsfiles er worden gemaakt, dit is:

$$\text{NSPLITS} = (\text{NKAN} + 4) / 10$$

waarbij: NSPLITS = aantal splitsfiles, naar boven afgerond op het eerstvolgende gehele getal

NKAN = aantal geregistreerde kanalen

Tijdens het splitsen van de registratiefile worden gelijktijdig nog enige controles uitgevoerd. Het programma vraagt het aantal kanalen per scan en het aantal kanalen per record, hieruit wordt berekend uit hoeveel records een scan bestaat en hoeveel splitsfiles er moeten worden gemaakt. Daarna wordt steeds een scan ingelezen en in het geheugen gezet. Ook worden dan de controles uitgevoerd. Deze controles bestaan achtereenvolgens uit:

1. Controle op alphanumerieke characters

Het programma verwacht alleen maar numerieke characters en zal een foutmelding geven wanneer het op een niet-numerieke character stuit.

2. Controle op klokregistratie

De klokregistratie, bestaande uit het dagnummer met het uur en de minuten, moet in een chronologische volgorde staan. Wanneer dit niet het geval is verschijnt er een foutmelding en men heeft binnen het programma de mogelijkheid tot het verbeteren van de fout.

3. Controle op proefnummer

Het proefnummer moet in de gehele registratiefile hetzelfde blijven, wanneer dit niet het geval is verschijnt er een foutmelding. Deze foutmelding kan een gevolg zijn van een tussentijds nieuw gekozen proefnummer, of van een andere recordindeling. Als er meer of minder kanalen worden geregistreerd kan het aantal records per scan veranderd zijn. Het programma leest dan in plaats van een proefnummer een kanaalnummer.

Nadat deze controles zijn uitgevoerd worden de registraties naar de splitsfiles weggeschreven. De splitsfiles dienen in een later stadium weer als input voor verdere verwerking van de gegevens.

Voorbeeld 4. Het splitsen van de registratiefile uit voorbeeld 1 met behulp van subroutine SPLITS

WAT WILT U DOEN :

- 1 = HET SPLITSEN VAN EEN REGISTRATIE-FILE
- 2 = HET TESTEN VAN DE METINGEN OP INTERVALLEN
- 3 = HET WISSELEN VAN KANALEN
- 4 = HET TRANSFORMEREN EN TABELLEREN
- 5 = STOPPEN

? 1

HOE HEET UW REGISTRATIE FILE ?VOORB.DEN
 HOEVEEL KANALEN PER RECORD ZIJN ER GEREGEREED? 7
 HOEVEEL KANALEN ZIJN ER PER SCAN GEREGEREED?23
 HOE HEET DE SPLITS-FILE
 ER ZIJN/WORDEN 3 SPLITS FILES GEMAAKT

?
 VOORB.SPL

E-O-F GELEZEN IN FILE 20

Voorbeeld 5. De in drie splitsfiles gesplitste file uit voorbeeld 1

-99999.99	23				20	21	22	23	24	25
4214	42	15	0	123	388	399	402	253	-32767	
4214	42	16	0	123	389	400	403	255	-32767	
4214	42	16	59	123	390	401	402	255	-32767	
30	31	32	33	34	35	1	2	3	7	
-32767	-32767	-32767	-32767	-32767	-32767	319	680	749	717	
126	332	310	210	-32767	-32767	319	681	749	705	
-32767	-32767	-32767	-32767	-32767	-32767	318	681	750	705	
8	10	11	12	14	15	16				
257	596	741	605	539	618	572				
256	597	742	607	541	618	579				
256	599	742	608	543	620	580				

Deze drie splitsfiles zijn gemaakt met behulp van subroutine SPLITS. Als registratiefile is de file uit voorbeeld 1 gebruikt. De sterretjes zijn eerst vervangen door de onbekend code -32767. Het eerste record van elke splitsfile bevat de kanaalnummers. De eerste splitsfile bevat ook een testgetal en het totaal aantal geregistreerde kanalen. Het tweede tot en met het vierde record bevat achtereenvolgens het proefnummer, de dagaanduiding, de klokregistratie en zes maal een meetwaarde. De tweede en derde splitsfile bevatten in het tweede tot en met vierde record alleen meetwaarden.

5.2. Het verbeteren van eenvoudige registratiefouten

Wanneer er tijdens het splitsen fouten in de registratiefile worden geregistreerd, is het mogelijk deze fouten te verbeteren, zonder dat men het programma moet stoppen. Dit kan met de subroutine BETTER. Wanneer er grove fouten in de registratiefile voorkomen is het raadzaam, het programma af te stoppen en eerst deze fouten met de editor te verbeteren. Als subroutine BETTER wordt aangeroepen dan wordt de pointer van de registratiefile op het begin van de scan gezet waarin de fout gevonden werd. Record voor record wordt de scan nu op het beeldscherm gezet en het programma vraagt wat er mee moet gebeuren. Er zijn 5 mogelijkheden waaruit men kan kiezen, te weten:

1. Het record blijft ongewijzigd
2. Het record wordt weggegooid
3. Voorafgaand aan dit record worden 1 of meer records tussengevoegd
4. Volgend op dit record worden 1 of meer records tussengevoegd
5. binnen dit record wordt een string met characters gewijzigd in een andere string met characters

De verbeterde scan wordt weggeschreven op een hulpfile. In subroutine SPLITS wordt nu deze hulp file gelezen en kan het splitsen van de registratiefile weer verder gaan. Als de gehele registratiefile gesplitst is, zal er een melding op het beeldscherm verschijnen dat er een 'end of file' gelezen is in file 20 (zie voorbeeld 4). Het is ook mogelijk dat er een melding verschijnt, dat er een end of file gelezen is in file 19. Dit betekent dat er in de hulpfile, die gemaakt is met de subroutine BETTER, een fout voorkomt. Dan zal alsnog de execu-

tie van het programma moeten worden gestopt en de fout in de registratiefile moeten worden verbeterd.

Voorbeeld 6. Een foute scan uit een registratiefile

```
04214,042 15 00/  
020 +0123,021 +0388,022 +0399,023 +0402,024 +0253,025 -32767,030 -32767/  
031 -32767,032 *****,033 -32767,034 -32767,035 -32767,001 +0319,002 +0680/  
003 +0749,007 +0717,008 +0257,010 +0596,011 +0741,012 +0605,014 +0539/  
015 +0618,016 +0572//
```

De fout bestaat hieruit dat in de registratiefile '*****' voorkomt in plaats van de onbekend code (-32767).

Voorbeeld 7. De scan uit voorbeeld 6 zoals die behoort te zijn

```
04214,042 15 00/  
020 +0123,021 +0388,022 +0399,023 +0402,024 +0253,025 -32767,030 -32767/  
031 -32767,032 -32767,033 -32767,034 -32767,035 -32767,001 +0319,002 +0680/  
003 +0749,007 +0717,008 +0257,010 +0596,011 +0741,012 +0605,014 +0539/  
015 +0618,016 +0572//
```

Bij kanaalnummer 32 moet dus '*****' worden vervangen door -32767. Voor het aanbrengen van de gewenste verbetering wordt subroutine BETTER opgeroepen.

Voorbeeld 8. Het verbeteren van de foute scan uit voorbeeld 6 met behulp van subroutine BETTER

```
031 -32767,032 *****,033 -32767,034 -32767,035 -32767,001 +0319,002 +0680/  
WILT U DIT RECORD VERBETEREN? JA  
WAT WILT U MET HET NA-VOLGENDE RECORD DOEN?  
1 = RECORD BLIJFT ONGEWIJZIGD  
2 = RECORD WORDT WEGGEGOOID  
3 = VOORAFGAAND AAN DIT RECORD, 1 OF MEER RECORDS TUSSENVOEGEN(AFSLUITEN MET  
4 = VOLGEND OP DIT RECORD, 1 OF MEER RECORDS TUSSENVOEGEN(AFSLUITEN MET })  
5 = BINNEN DIT RECORD WORDT ER VERBETERD
```

```
04214,042 15 00/  
WELK VAN DE 5 MOGELIJKHEDEN WENST U VOOR DIT RECORD TE GEBRUIKEN ?1  
020 +0123,021 +0388,022 +0399,023 +0402,024 +0253,025 -32767,030 -32767/  
WELK VAN DE 5 MOGELIJKHEDEN WENST U VOOR DIT RECORD TE GEBRUIKEN ?1  
031 -32767,032 *****,033 -32767,034 -32767,035 -32767,001 +0319,002 +0680/  
WELK VAN DE 5 MOGELIJKHEDEN WENST U VOOR DIT RECORD TE GEBRUIKEN ?5  
TIK IN OP DE MANIER VAN .....=.....  
(= IS VERPLICHT ALS SCHEIDINGSTEKEN)
```

```
*****=-32767  
003 +0749,007 +0717,008 +0257,010 +0596,011 +0741,012 +0605,014 +0539/  
WELK VAN DE 5 MOGELIJKHEDEN WENST U VOOR DIT RECORD TE GEBRUIKEN ?1  
015 +0618,016 +0572//  
WELK VAN DE 5 MOGELIJKHEDEN WENST U VOOR DIT RECORD TE GEBRUIKEN ?1  
E-O-F GELEZEN IN FILE 20
```


De gehele scan wordt dus met behulp van subroutine BETTER behandeld. Alle records blijven ongewijzigd behalve het derde record waarin '*****' veranderd wordt in -32767. Nadat de gehele scan behandeld is gaat de subroutine SPLITS weer verder met het splitsen van de registratiefile. Wanneer het programma hiermee klaar is geeft het de melding: E-O-F GELEZEN IN FILE 20.

5.3. Het wisselen van kanalen

Het is mogelijk om bij de datalogger de metingen van twee of meer kanalen onderling te verwisselen. Dit heeft dan gevolgen voor het ijken van de metingen met behulp van ijkformules. De metingen van de verwisselde kanalen vormen dan geen logische reeks meer. De metingen moeten dan eerst weer in een logische reeks worden geplaatst. Dit kan met behulp van de subroutine WISSEL gebeuren. De splitsfiles dienen nu als inputfiles voor het programma LOGGER. Het inlezen en weer wegschrijven van de metingen van en naar de splitsfiles gebeurt met behulp van de subroutine HULP. Na het wisselen van de kanalen zijn er eenzelfde aantal splitsfiles bijgemaakt. Daar de oude splitsfiles nu overbodig zijn bestaat de mogelijkheid om deze splitsfiles vanuit de subroutine HULP van de schijf te verwijderen. De input die subroutine WISSEL nodig heeft bestaat achtereenvolgens uit:

1. KANAALNUMMER
2. KANAALNUMMER
3. BEGINDAG
4. BEGINUUR

Voorbeeld 9. Wisselen van kanalen, welke in de tweede splitsfile van voorbeeld 5 voorkomen

```
HOE HEET UW EERSTE SPLITS-FILE? VOORB.SPL#1
GEEF NU DE KANALEN OP DIE VERWISSELD MOETEN WORDEN
AFSLUITEN MET 9999
WANNEER VAN AF HET BEGIN MOET WORDEN VERWISSELD VOOR DAG EEN 0 INTIKKEN
KANAALNR :1
VERWISSELEN MET KANAALNR : 35
BEGINDAG :0
KANAALNR :9999
HOE HETEN DE NIEUWE SPLITS-FILES ? VSPLIT.SPL
WILT U DE OUDE SPLITS-FILES BEWAREN ?JA
```

Voorbeeld 10. Splitsfile na het wisselen van kanalen

```
>PIP TI:=VSPLIT,SPL;2
 30   31   32   33   34   35   1   2   3   7
-32767 -32767 -32767 -32767 -32767 319 -32767 680 749 717
 126   332   310   210 -32767 319 -32767 681 749 705
-32767 -32767 -32767 -32767 -32767 318 -32767 681 750 705
```

5.4. Het testen van de metingen op intervallen

Een datalogger kan foute meetwaarden registreren. Meestal kan men wel vrij nauwkeurig schatten in welke orde van grootte een meetwaarde behoort te liggen. Men kan dus een interval opstellen, waarin de meetwaarden zouden moeten vallen, waarvan men de juistheid betwijfelt. Deze meetwaarden zouden dan vervangen moeten worden door andere meetwaarden of onbekend codes. Dit aspect is mogelijk gemaakt met behulp van de subroutine INTVAL. Ook hier zorgt subroutine HULP weer voor het inlezen en wegschrijven van en naar de splitsfiles. Eveneens worden er weer nieuwe splitsfiles gemaakt. De oude splitsfiles kunnen, evenals bij het wisselen van kanalen het geval is, weer uit het systeem worden verwijderd. De input voor subroutine INTVAL bestaat achtereenvolgens uit:

1. KANAALNUMMER
2. ONDERGRENS
3. BOVENGRENS
4. VERVANGENDE WAARDE

Wanneer deze vier waarden zijn ingelezen vraagt de subroutine opnieuw een kanaalnummer. Men kan dus verschillende intervallen opgeven. Als men geen verdere intervallen meer wil opgeven moet in plaats van het kanaalnummer een negatief getal met een slash worden ingetypt (bijvoorbeeld -1/). Ook kan men alle kanalen tegelijk testen op eenzelfde interval. Dan moet men voor het kanaalnummer 9999 intypen. Er zijn bovendien twee mogelijkheden om de input voor de subroutine INTVAL in te lezen. Deze twee mogelijkheden zijn:

1. Vanaf het toetsenbord van de terminal
2. Vanaf een file, die op schijf staat

De subroutine vraagt dan ook waar vandaan de inputgegevens gelezen moeten worden. Als het toetsenbord gewenst wordt moet voor de filenaam TI: ingetypt worden terwijl in het andere geval een file specificatie van de file waar de gegevens opstaan moet worden opgegeven.

Voorbeeld 11. Het testen op intervalwaarden van de splitsfiles uit voorbeeld 5 met behulp van subroutine INTVAL

```
HOE HEET UW EERSTE SPLITS-FILE? VOORB.SPL:1
SUBROUTINE INTVAL VOOR HET TESTEN OP INTERVAL WAARDEN
DE INPUT VOOR DEZE SUBROUTINE DIENST ALS VOLGT TE WORDEN OPgegeven ;
KANAALNUMMER ONDERGRENs BOVENGRENs VERVANGENDE-WAARDE

WANNEER VOOR ALLE KANALEN OP EEN ZELFDE INTERVAL GETEST
MOET WORDEN , DIENST U IN PLAATS VAN HET KANAALNR. HET GETAL 9999 IN TE VOEREN.
WANNEER U IN PLAATS VAN HET KANAALNR EEN NEGATIEF GETAL MET EEN /
INVOERT BETEKENT DIT EINDE INPUT

VANAF WELKE FILE WORDEN DE INPUTGEGEVENS GELEZEN?
INGEVAL U VANAF DE TERMINAL DE GEGEVENS IN WILT LEZEN
MOET U VOOR FILE-NAAM : TI: INVULLEN
FILE-NAAM =TI:
TIK IN: KANAALNR. ONDER- EN BOVENGRENs EN VERVANGENDE WAARDE
OF      ; EEN NEGATIEF GETAL MET EEN / VOOR EINDE INVOER
INVOER =

22 400 500 1000
TIK IN: KANAALNR. ONDER- EN BOVENGRENs EN VERVANGENDE WAARDE
OF      ; EEN NEGATIEF GETAL MET EEN / VOOR EINDE INVOER
INVOER =

15 600 700 0
TIK IN: KANAALNR. ONDER- EN BOVENGRENs EN VERVANGENDE WAARDE
OF      ; EEN NEGATIEF GETAL MET EEN / VOOR EINDE INVOER
INVOER =

-1/
HOE HETEN DE NIEUWE SPLITS-FILES ? VNEW.SPL
WILT U DE OUDE SPLITS-FILES BEWAREN ?JA
```

Uit voorbeeld 11 blijkt dat alle gegevens worden gelezen vanaf de terminal. De kanalen 22 en 15 worden op intervallen getest. Wanneer de meetwaarden van kanaal 22 in het interval 400; 500 ligt wordt deze meetwaarde vervangen door 1000 eveneens is dit het geval bij kanaal 15, doch hier bestaat het interval uit 600; 700 en is de vervangende

waarde 0. De invoer van de gegevens wordt afgesloten door in plaats van een volgende kanaalnummer -1/in te typen. De nieuwe meetwaarden worden weggeschreven op nieuwe splitsfiles, namelijk VNEW.SPL. Deze splitsfiles zijn afgebeeld in voorbeeld 12. De oude splitsfiles worden hier bewaard.

Voorbeeld 12. Splitsfiles uit voorbeeld 5 na het testen op intervalwaarden

-99999.99	23			20	21	22	23	24	25
4214	42	15	0	123	388	399	402	253	-32767
4214	42	16	0	123	389	1000	403	255	-32767
4214	42	16	59	123	390	1000	402	255	-32767
30	31	32	33	34	35	1	2	3	7
-32767	-32767	-32767	-32767	-32767	-32767	319	680	749	717
126	332	310	210	-32767	-32767	319	681	749	705
-32767	-32767	-32767	-32767	-32767	-32767	318	681	750	705
8	10	11	12	14	15	16			
257	596	741	605	539	0	572			
256	597	742	607	541	0	579			
256	599	742	608	543	0	580			

5.5. Het transformeren en tabelleren van de meetwaarden

De meetwaarden van een datalogger zijn vaak een sommatie van een aantal impulsen per tijdsinterval. Om deze impulsen om te zetten in de gewenste eenheden wordt er van ijk- of transformatieformules gebruik gemaakt. Definitieve meetwaarden worden verkregen door toepassing van deze formules. De nieuwe getransformeerde meetwaarden kunnen dan worden getabelleerd. Deze subroutines zijn voor elke datalogger weer anders. Men kan zelf een subroutine schrijven en deze aan het programma koppelen. Hoe dit in zijn werk gaat zal in par. 6 worden toegelicht. Voorbeeld 13 laat zien hoe een transformatie en een tabel-leringssubroutine zou kunnen werken.

Voorbeeld 13. De werking en het resultaat van een transformatie en tabellerings subroutine met behulp subroutine YK

TIK IN VOOR DATALOGGER:
 REMOTE SENSING EEN 1
 BORSSELE EEN 2

?
 2
 WILT U 1 WAARNEMING PER ETMAAL ? NEE
 HOE HEET UW EERSTE SPLITS-FILE? VOORB.SPL:1
 WILT U DE OUDE SPLITS-FILES BEWAREN ?NEE
 WILT U 1 WAARNEMING PER ETMAAL ? NEE

REGISTRATIEVERWERKING MEET-OPSTELLING BORSSELE EERSTE TABEL			NEERSLAG			AFVOERSNELHEID				BELEIDINGSVERMOELEN		DATALOGER BORSSELE		12-NOV-81 RECORDNUMMER		
DAO	UUR	MIN	METER 0.1MM	/SCAN 0.1MM	/DAB 0.1MM	DRAIN 1 ML/S	2 ML/S	1 MUM/D	2 MUM/D	M8 MV	CL/L	IN MV	VOCHT	UREN OF RECORDS	TEKEN VERSPRINGEN	IN DE DATA -FILE
42	15	0	35	0	*	17		379		319	3190					1
42	14	0	35	0	*	17		379		319	3190					2
42	14	59	35	0	*	17		379		318	3180					3

6. HET AANKOPPELEN VAN EEN NIEUWE TRANSFORMATIE EN TABELLERINGS-SUBROUTINE

Elke datalogger heeft voor de meetwaarden de eigen specifieke ijkingsformules. Voor de verwerking van de gegevens van een nieuwe datalogger, zal ook een nieuwe ijkings subroutine geschreven moeten worden. Deze subroutine moet dan op eenvoudige wijze aan het reeds bestaande programma gekoppeld kunnen worden. Bij de programmering van deze subroutine moet dan wel op een aantal punten worden gelet. Deze punten zijn:

1. Bij de declaratieve statements moet voorkomen:

```
REAL MEET
```

```
COMMON MEET(100),MEETW(100),KAN(100)
```

2. De reeks MEETW(KAN(I)) bevat de metingen, met als indices de kanaalnummers

De reeks KAN(I) bevat de kanaalnummers met als indices 1 tot NKAN

De reeks MEET(1) bevat geen waarden, maar is een reserve array voor eventuele conversie

3. De aanroep van de subroutine wordt in subroutine HULP verzorgt. Als argumenten bij de aanroep zouden kunnen worden meegegeven:

- a. NKAN = aantal geregistreerde kanalen
- b. NSPLTS = aantal spitfiles
- c. PROEF = proefnummer
- d. NDAG = dagnummer
- e. NUUR = uur
- f. MIN = minuten

Het aankoppelen van een nieuwe subroutine voor het ijken en tabelleren wordt in de subroutine HULP verzorgd. De volgende statements zijn daarbij van belang:

1. Statementnummer 12

Dit is een format waarin de beschikbare ijksubroutines worden genoemd. Voor een nieuwe subroutine moet de mogelijkheid van beschikbare subroutines met de nieuwe subroutine worden uitgebreid. Wanneer voor datalogger X een nieuwe subroutine is geschreven zou het format er als volgt uit kunnen zien:

```
3 TYPE 102
102 FORMAT(' TIK IN VOOR DATALOGGER: '/
*      ' REMOTE SENSING EEN 1' /
*      ' BORSSELE      EEN 2' /
*      ' NIEUWE X      EEN 3' /)
ACCEPT *, ILOG
```

2. Statementnummer 14

IF(ILOG.LT.1.OR.ILOG.GT.n) n moet gelijk zijn aan het aantal loggers genoemd in statementnummer 12. Met de nieuwe logger X erbij zou het statement er als volgt uitzien:

```
ACCEPT *, ILOG
IF(ILOG.LT.1.OR.ILOG.GT.3) GOTO 3
IF(ILOG.NE.2) GOTO 26
```

3. Statementnummer 111

De computed goto statement moet met een opvolgend nummer worden uitgebreid. Voor de nieuwe logger X is dit:

C

```
GOTO(60,61,62), ILOG
```

WELKE LOGGER?

C

4. Voor het statement met:

```
14 GOTO 7
```

Voor dit statement kan de tabel outputfile worden klaar gezet en kan ook de nieuwe ijkingssubroutine aangeroepen worden.

De statements die dan tussengevoegd moeten worden zijn:

```
NDAG1=0
:
GOTO 7
62 IF(IROUT.EQ.3) OPEN(UNIT=7,NAME='TABEL.OUT',TYPE='NEW')
CALL NIEUWX(NKAN,NSPLTS,PROEF,NDAG,NUUR,MIN)
IROUT=0
:
14 GOTO 7
```

TABEL.OUT = Nieuwe filenaam, waarop de getabelleerde gegevens geschreven kunnen worden

NIEUWX = nieuwe subroutine

IROUT = test getal

UNIT = logical unitnummer (te gebruiken zijn: 6, 7, 8 of 9)

De subroutine HULP en de nieuwe subroutine voor het ijken en het tabelleren moeten worden vertaald. Dit kan dan met de compiler F4P.

Dit kan als volgt:

```
F4P HULP,HULP=HULP/NOSP
```

```
F4P NIEUW,NIEUW=NIEUW/NOSP
```

Hierna moet met de taskbuilder van het PDP 11/70 operating systeem het programma met de subroutines opnieuw worden 'gelinked'. Dit kan als volgt:

```
TKB
```

```
TKB>RREGIS/FP=LOGGER,HULP,YK1,YK2,NIEUW,ACHT,WISSEL,INTVAL,DRO:[1,1]F4POTS/LB
```

```
TKB>/
```

```
ENTER OPTIONS:
```

```
TKB>LIBR=F4PRES:RO
```

```
TKB>UNITS=23
```

```
TKB>ACTFIL=10
```

```
TKB>ASG=SY:6
```

```
TKB>//
```

7. OVERGANG NAAR HET PAKKET 'VAZAL'

De mogelijkheid bestaat om de verkregen data te converteren naar binaire getallen en deze te plaatsen in een datafile, zoals deze wordt voorgesteld in het pakket 'VAZAL'. Verdere verwerking van de data is dan mogelijk (b.v.: plotten, kopiëren, transformeren van data). Het overbrengen van de data naar een datafile kan met behulp van het programma INPUT uit het pakket 'VAZAL'. Hoe dit programma kan worden gebruikt kan men vinden in de programmabeschrijving van VAN GILS (1981) voor het conversationeel gebruik op de PDP 11/70 te Utrecht. Er dient wel rekening gehouden te worden met het feit, dat het programma LOGGER de getabelleerde data wegschrijft met de tabelopschriften erbij. Afhankelijk van welke datalogger men de outputfile wil converteren, moet rekening worden gehouden met het aantal tabelopschriftregels dat moet worden overgeslagen (skippen). Dat deze regels moeten worden geskipt kan men ook in programma INPUT opgeven.

8. BIJLAGEN

De eerste bijlage is een schematische voorstelling van de datastroom. De tweede bijlagen zijn de afspraken die gemaakt zijn omtrent het registratieformat van de dataloggers. Verder zal in de volgende bijlagen de volledige tekst van de ontwikkelde programma's worden gegeven. De bijlagen zijn achtereenvolgens:

- Bijlage 1. Schematische voorstelling van datastroom
- Bijlage 2. Afspraken die gemaakt zijn tussen de Stuurgroep Programma-ontwikkeling Automatische Registratieverwerking en de Technische en Fysische Dienst voor de Landbouw omtrent het registratieformat van de dataloggers
- Bijlage 3. Programma LOGGER. Het hoofdprogramma zorgt voor het oproepen van de gewenste subroutines. De werking van het hoofdprogramma is beschreven in de aanvang van hoofdstuk 5
- Bijlage 4. Subroutine SPLITS. Deze subroutine zorgt voor het splitsen van de registratiefile. De beschrijving van subroutine splits staat in hoofdstuk 5.1. Met deze subroutine zijn de voorbeelden 4 en 5 verwerkt

- Bijlage 5. Subroutine BETTER. Deze subroutine zorgt voor het verbeteren van eenvoudige registratiefouten. De beschrijving van deze subroutine staat in hoofdstuk 5.2. Met deze subroutine zijn de voorbeelden 6, 7 en 8 verwerkt
- Bijlage 6. Subroutine HULP. Deze subroutine zorgt voor het lezen en wegschrijven van en naar de splitsfiles. Deze subroutine kan niet door de gebruiker worden aangeroepen. Het programma roept deze subroutine automatisch aan, wanneer deze nodig is
- Bijlage 7. Subroutine WISSEL. Deze subroutine zorgt voor het wisselen van de kanalen. De beschrijving van deze subroutine staat in hoofdstuk 5.3. Met deze subroutine zijn de voorbeelden 9 en 10 verwerkt
- Bijlage 8. Subroutine INTVAL. Deze subroutine zorgt voor het testen op intervalwaarden. De beschrijving van deze subroutine staat in hoofdstuk 5.4. Met deze subroutine zijn de voorbeelden 11 en 12 verwerkt
- Bijlage 9. Subroutine YK2. Deze subroutine zorgt voor het ijken en tabelleren van de meetwaarden. De beschrijving van deze subroutine staat in hoofdstuk 5.5. Met deze subroutine is voorbeeld 13 verwerkt

REFERENTIES

- GILS, J.B.H.M. VAN, 1981. Programmapakket 'VAZAL'. Beschrijving op de PDP 11/70 van het TWR in Utrecht.

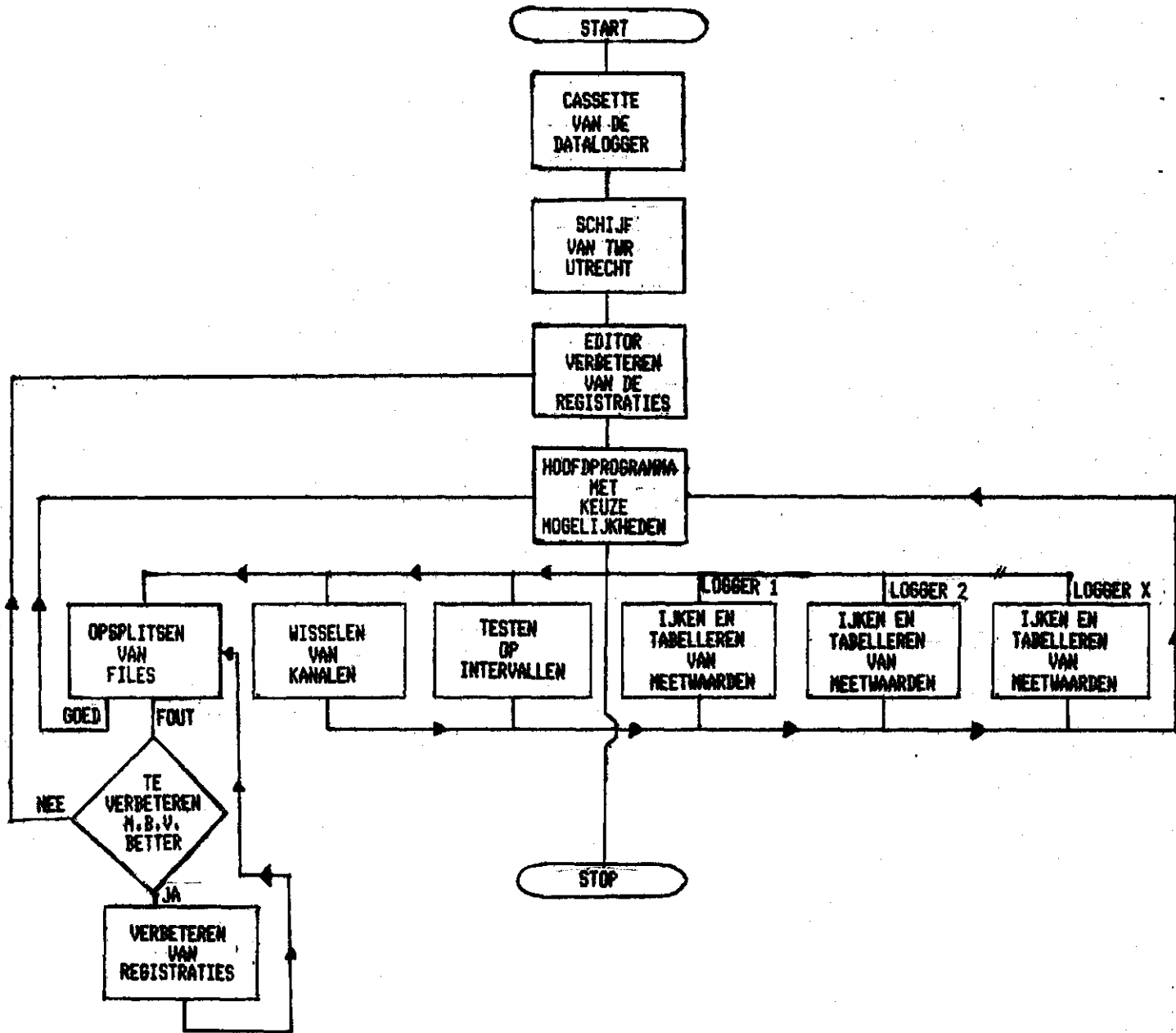
Bijlage 1.

SCHEMATISCHE VOORSTELLING

VAN DE

DATASTROOM

Bijlage 1. Schematische voorstelling van de datastroom



Bijlage 2.

**AFSPRAKEN
OVER HET
REGISTRATIEFORMAT**

GEMAAKTE AFSPRAKEN MET DE TFDL OVER HET REGISTRATIEFORMAT VAN EEN
DATALOGGER

Een scan van een datafile behoort er als volgt uit te zien:

- Het beginrecord moet achtereenvolgens bevatten:

4 spaces als beginaanduiding van een scan
xxxxx voor het proefnummer van de cassette
 , als scheidingsteken tussen proefnummer en klokregistratie
xxx voor het dagnummer
space als scheidingsteken
 xx voor het uur
space als scheidingsteken
 xx voor de minuten
/space voor het einde van het beginrecord en de klokregistratie
CrLf voor carriage return, line feed

- Ieder volgend record van de scan bevat de kanaalnummers met hun
meetwaarden. De records moeten eenzelfde recordlengte hebben, met
uitzondering van het laatste record van de scan, welke kleiner kan
zijn dan de voorgaande records. De indeling van de kanalen records
is dan:

n maal xxx voor het kanaalnummer
 space als scheidingsteken tussen kanaalnummer en meting
 xxxxx voor de meting
 , voor de kanaalscheiding
n is het aantal kanalen dat per record geregistreerd wordt
 / of // voor respectievelijk einde van een record of
 einde van een scan
 CrLf voor een carriage return line feed

- Bij stroominschakeling moet eerst door de logger STARTCrLfSTARTCrLf
worden geschreven.

Bijlage 3.

HOOFDPROGRAMMA

LOGGER

```
C
C-----
C
C          PROGRAMMA LOGGER                K.OOSTINDIE  ICW
C
C PROGRAMMA LOGGER VOOR HET VERWERKEN VAN AUTOMATISCH GEREGBSTREERDE GEGEVENS
C DIT PAKKET BESTAAT UI EEN HOOFDPROGRAMMA MET VERSCHILLENDE SUBROUTINE'S
C
C HET HOOFDPROGRAMMA ZORGT VOOR HET OPROEPEN VAN DE VERSCHILLENDE SUBROUTINE'S
C
C DE SUBROUTINE'S DIE DOOR HET HOOFDPROGRAMMA OPGEROEPEN WORDEN ZIJN :
C
C          ---SPLITS---HULP---INTVAL---BETTER---WISSEL---
C
C          EN EEN DOOR U ZELF GESCHREVEN YKINGS EN TABELLERINGSSUBROUTINE
C-----
C
```

```
0001      BYTE NFILE(35),ANS,DUNNY
0002          REAL MEET
0003          COMMON MEET(100),MEETW(100),KAN(100)
```

```
0004      102 FORMAT(' HOE HEET UW REGISTRATIE FILE ?'*)
0005      GOTO 3
```

```
C-----
C          OPRAGEN VAN GEGEVENS VOOR HET SPLITSEN VAN DE REGISTRATIEFILE
C-----
C
```

```
0006      8 TYPE 102
0007          ACCEPT 103,NFILE
0008          NFILE(35)=0
0009          OPEN(UNIT=20,NAME=NFILE,TYPE='OLD',ERR=8)
0010          OPEN(UNIT=19)
0011      103 FORMAT(35A1)
0012      104 FORMAT(' HOEVEEL KANALEN PER RECORD ZIJN ER GEREGBSTREERD? '*)
0013          TYPE 104
0014          ACCEPT *,KANREC
0015      1 TYPE 105
0016      105 FORMAT(' HOEVEEL KANALEN ZIJN ER PER SCAN GEREGBSTREERD?')
0017          ACCEPT *,NKAN
0018          NSPLIT=(NKAN+4.)/10.+999
0019          TYPE 106,NSPLIT
0020          ACCEPT 103,NFILE
0021          NFILE(35)=0
0022          DO 2 I=1,NSPLIT
0023              J=I+10
0024          2 OPEN(UNIT=J,NAME=NFILE,TYPE='NEW')
0025              CALL SPLITS(KANREC,NKAN,NSPLIT)
0026              CLOSE(UNIT=20,DISP='KEEP')
0027              CLOSE(UNIT=19,DISP='DELETE')
```

```
0028      106 FORMAT(' HOE HEET DE SPLITS-FILE'' ER ZIJN/WORDEN '12
*' SPLITS FILES GEMAAKT'' ?')
```

```
0029      107 FORMAT(' WAT WILT U DOEN ?')
*' 1 = HET SPLITSEN VAN EEN REGISTRATIE-FILE'/
*' 2 = HET TESTEN VAN DE METINGEN OP INTERVALLEN'/
```

*' 3 = HET WISSELEN VAN KANALEN'/'
*' 4 = HET TRANSFORMEREN EN TABELLEREN '/'
*' 5 = STOPPEN '/' ? '€)

C-----
C KEUZEMOGELIJKHEDEN VOOR DE SUBROUTINE'S
C-----

0030 3 TYPE 107
0031 ACCEPT *,K
0032 IF(K.LT.1.OR.K.GT.5) GOTO 3

C-----
C AANROEPEN SUBROUTINE'S
C-----

0033 GOTO(8,4,5,6,7),K
0034 4 CONTINUE
0035 CALL HULP(4)
0036 GOTO 3
0037 5 CONTINUE
0038 CALL HULP(2)
0039 GOTO 3
0040 6 IROUT=3
0041 CALL HULP(IROUT)
0042 GOTO 3
0043 7 STOP
0044 END

Figure 4.

SUBROUTINE

SPLITS

```
0001      SUBROUTINE SPLITS(KANREC,NKAN,NSPLIT)
C
C-----
C
C      SUBROUTINE SPLITS                K.OOSTINDIE ICM
C
C      SUBROUTINE SPLITS ZORGT VOOR HET SPLITSEN VAN DE REGISTRATIEFILE
C      VAN EEN DATALOGGER. IN DEZE ROUTINE WORDEN OOK DE MOGELIJKE CONTROLE'S
C      UITGEVOERD OP HET JUIST REGISTREREN VAN DE DATALOGGER. WANNEER DE ROUTINE
C      EEN FOUT ONTDEKT , HEEFT MEN DE MOGELIJKHEID DEZE TE VERBETEREN.
C      SUBROUTINE BETTER WORDT DAN AANGEROEPEN.
C
C-----
0002      DIMENSION NHULP(9),KAN1(100)
0003          REAL MEET
0004          COMMON MEET(100),MEETW(100),KAN(100)
0005      INTEGER PROEF,UNIT
0006      BYTE ANS,REEKS(80),NFILE(35)
C
0007      NVOR=0
0008      UNIT=20
0009      KANWR=1
0010      NPLUS=KANREC
0011      ANKAN=NKAN
0012      AANREC=KANREC
0013      MREC=ANKAN/AANREC+1.99
0014      N1=1
0015      ITEST=0
0016      JTEST=0
0017      NDAG=0
0018      NREC=1
C
0019      1 KTEST=0
0020      2 MREC1=0
0021      MREC1=MREC1+1
C-----
C
C
C-----
C
C      KLOKREGISTRATIE
C-----
0022      READ(UNIT,*,END=999,ERR=11) PROEF,NDAG,NUUR,MIN
0023      3 KTEST=0
0024      ITEST=0
0025      MREC1=MREC1+1
C-----
C
C
C-----
C
C      KANALEN MET MEETWAARDEN
C-----
0026      READ(UNIT,*,END=999,ERR=11) (KAN(I),MEETW(I),I=N1,NPLUS),ITEST
0027      IF(ITEST.NE.0) GOTO 7
0028      18 IF(NPLUS.EQ.NKAN) GOTO 4
0029      NPLUS=NPLUS+KANREC
0030      N1=N1+KANREC
0031      IF(NPLUS.GE.NKAN) NPLUS=NKAN
0032      GOTO 3
C-----
C
C      CONTROLE OF DE KANALEN IN DEZELFDE VOLGORDE
C      BEREGBSTREERD WORDEN
C
```

```
-----  
C  
0033 4 UNIT=20  
0034 DO 10 I=1,NKAN  
0035 IF(NREC.EQ.1.OR.JTEST.EQ.1) GOTO 14  
0036 IF(KAN(I).NE.KAN1(I).AND.(KAN1(I).NE.-32767.AND.KAN(I).NE.-32767))  
*GOTO 9  
0037 GOTO 14  
-----  
C KEUZEGELIJKHEDEN WANNEER DE KANALEN NIET IN DEZELFDE VOLGORDE STAAN  
-----  
C  
0038 9 TYPE 100  
0039 TYPE *,PROEF,NDAG,NUUR,MIN  
0040 TYPE *,(KAN(K),MEETW(K),K=1,NKAN)  
0041 ACCEPT *,IG  
0042 GOTO(16,19,15),IG  
C  
0043 16 CALL BETTER(NREC,UNIT,NREC1) SUBROUTINE BETTER  
0044 NPLUS=KANREC  
0045 N1=1  
0046 GOTO 1  
0047 14 KAN1(I)=KAN(I)  
C  
0048 10 CONTINUE  
0049 JTEST=0  
0050 IF(NREC.EQ.1) GOTO 5  
-----  
C CONTROLE OF HET PROEFNUMMER BELIJK BLIJFT EN OF DE KLOKREGISTRATIE IN EEN  
C CHRONOLOGISCHE VOLGORDE STAAT  
-----  
C  
0051 IF(KONREC.NE.PROEF) GOTO 13  
0052 5 IF((NDAG.LT.NV.OR.NUUR.GT.23.OR.MIN.GT.59).AND.(NDAG.NE.-32767.OR.  
*NV.NE.-32767)) GOTO 8  
0053 GOTO 24  
C  
0054 15 STOP  
C  
0055 100 FORMAT(' ER KOMT EEN FOUT IN DE KANALEN VOOR'/  
*' TIK IN : EEN 1 VOOR VERBETEREN'/  
*' 2 VOOR ANDERE KANAAL-INDELING '/  
*' 3 VOOR STOP '*)  
C  
0056 24 NDAG1=NDAG  
0057 KONREC=PROEF  
0058 N1=1  
0059 NPLUS=6  
-----  
C KANALEN MET MEETWAARDEN WEGSCHRIJVEN  
-----  
C  
0060 DO 20 I=1,NBPLIT  
0061 K=I+10  
0062 IF(I.EQ.1) GOTO 31  
0063 IF(KANWR.EQ.1) WRITE(K,*) (KAN(J),J=N1,NPLUS)  
0064 IF(NPLUS.GE.NKAN) KANWR=0  
0065 WRITE(K,*) (MEETW(J),J=N1,NPLUS)  
0066 GOTO 32  
0067 31 IF(KANWR.EQ.1) WRITE(K,200) NKAN,(KAN(J),J=N1,NPLUS)
```

```
0068 WRITE(K,*) PROEF,NDAG,NUUR,MIN,(MEETW(J),J=N1,NPLUS)
0069 32 N1=NPLUS+1
0070 NPLUS=NPLUS+10
0071 20 IF(NPLUS.GT.NKAN) NPLUS=NKAN
0072 200 FORMAT(1H,'-99999.99' I2,13X,6I7)
0073 KANWR=0
0074 NPLUS=KANREC
0075 N1=1
0076 NREC=NREC+1
0077 DO 50 I=1,NKAN
0078 MEETW(I)=0
0079 50 KAN(I)=0
0080 GOTO 1
```

C-----
C AANTAL KANALEN PER SCAN KLOPT NIET
C-----

```
0081 7 KTEST=1
0082 UNIT=20
0083 TYPE 101
0084 101 FORMAT(' AANTAL KANALEN PER SCAN KLOPT NIET')
0085 TYPE 116,NDAG,NUUR,MIN,NREC
0086 116 FORMAT(' KLOK REG. = '3I6/' IN SCANNR. 'I6)
0087 GOTO 11
```

C-----
C FOUTE KLOKREGISTRATIE
C-----

```
0088 8 TYPE 102,NDAG,NUUR,MIN,NDAG1
0089 102 FORMAT(' FOUT IN KLOKREGISTRATIE'/
* DAGNR= 'I4/
* UUR= 'I4/
* MIN= 'I4/
* VORIG DAGNR= 'I4//
* WILT U DIT VERBETERENT? '$)
0090 ACCEPT 103,ANS
0091 103 FORMAT(132A1)
0092 IF(ANS.EQ.'J') CALL BETTER(MREC,UNIT,MREC1)
0093 NPLUS=KANREC
0094 N1=1
0095 GOTO 1
```

```
C
0096 11 BACKSPACE 20
0097 READ(20,103) REEKS
0098 TYPE 201,REEKS
0099 201 FORMAT(1H,'80A1)
0100 TYPE 104
0101 104 FORMAT(' WILT U DIT RECORD VERBETERENT? '$)
0102 ACCEPT 103,ANS
0103 IF(ANS.NE.'J') GOTO 6
0104 CALL BETTER(MREC,UNIT,MREC1)
0105 NPLUS=KANREC
0106 N1=1
0107 GOTO 1
```

C-----
C KEUZE MOGELIJKHEDEN INDIEN ER NIET MOET WORDEN VERBETERD
C-----

```
0108      6 IF(KTEST.EQ.1) GOTO 12
0109      TYPE 105
0110      105 FORMAT(' WAT WILT U DAN: '//
      *' EEN 1 VOOR VERGEET RECORD EN MAAK ONBEKEND-KODE '//
      *' 2 VOOR VERGEET RECORD ER VOLGT EEN ANDER IN DE REG-FILE '//
      *' 3 VOOR STOP '//
0111      ACCEPT *,IG
0112      IF(NREC1.EQ.0) GOTO(17,2,15),IG
0113      GOTO(17,18,15),IG
C
0114      12 TYPE 106
0115      106 FORMAT(' WAT WILT U DAN : '//
      *' EEN 1 VOOR WAT NA HET LAATSTE KANAAL KOMT KAN VERGETEN WORDEN '//
      *' 2 VOOR EEN ANDERE RECORD-INDELING '//
      *' 3 VOOR STOP '//
0116      ACCEPT *,IG
0117      GOTO(18,19,15),IG
C-----
C                                RECORD VAN DE SCAN WORDT ONBEKEND GEMAAKT
C-----
0118      17 IF(NREC.EQ.1) GOTO 21
0119      DO 30 I=N1,NPLUS
0120      KAN(I)=-32767
0121      30 MEETW(I)=-32767
0122      GOTO 18
C
0123      21 PROEF=-32767
0124      NDAG=-32767
0125      NUUR=-32767
0126      MIN=-32767
0127      GOTO 3
C-----
C                                NIEUWE KANAALINDELING
C-----
0128      19 TYPE 107
0129      KANWR=1
0130      107 FORMAT(' HOEVEEL KANALEN WORDEN GEREgistREERDT?')
0131      NKAN1=NKAN
0132      ACCEPT *,NKAN
0133      TYPE 108
0134      108 FORMAT(' HOEVEEL KANALEN PER RECORD ZIJN ER?')
0135      IF(NVOR.LT.NSPLIT) NVOR=NSPLIT
0136      KANR=KANREC
0137      ACCEPT *,KANREC
0138      KBPLIT=(NKAN+4.)/10.+999-NVOR
0139      IF(NKAN1.GE.NKAN.OR.PROEF.EQ.KONREC) GOTO 61
0140      DO 62 I=1,NSPLIT
0141      J=I+10
0142      62 BACKSPACE J
0143      MREC1=MREC1+MREC
0144      61 IF(KBPLIT.LE.0) GOTO 27
0145      DO 40 I=1,KBPLIT
0146      J=I+NVOR
0147      TYPE 113,J
0148      J=J+10
0149      ACCEPT 103,NFILE
```

LOGGER.FTN)12 /TR:BLOCKS/WR

```

0150          NFILE(35)=0
0151      40 OPEN(UNIT=J,NAME=NFILE,TYPE='OLD')
0152          NSPLIT=NSPLIT+KSPLIT
0153      113 FORMAT(' HOE HEET SPLITSFILE NR. 'I2' ? '$)
0154      27 MREC=(NKAN/KANREC+.99)+1.
0155          NSPLIT=(NKAN+4.)/10.+999
0156          IF(NKAN.NE.(NKAN/KANREC)*KANREC) MREC=MREC+1
0157          DO 60 I=1,MREC1
0158      60 BACKSPACE 20
0159          NPLUS=KANREC
0160          N1=1
0161          JTEST=1
0162          GOTO 1

```

```

C-----
C                                     KEUZEMOGELIJKHEDEN BIJ EEN FOUT PROEFNUMMER
C-----

```

```

0163      13 TYPE 112
0164      112 FORMAT(' FOUT IN PROEFNUMMER '$)
0165          TYPE *,PROEF,NDAG,NUUR,MIN
0166          IF(NREC.EQ.2) GOTO 22
0167          TYPE 109
0168      109 FORMAT(' IS UW RECORD-INDELING VERANDERD ? '$)
0169          ACCEPT 103,ANS
0170          IF(ANS.EQ.'J') GOTO 19
0171          TYPE 110
0172      110 FORMAT(' MOET DIT RECORD DAN VERBETEN WORDEN ? '$)
0173          ACCEPT 103,ANS
0174          IF(ANS.EQ.'J') GOTO 2
0175          TYPE 114
0176      114 FORMAT(' IS DIT EEN NIEUW PROEFNUMMER ? '$)
0177          ACCEPT 103,ANS
0178          IF(ANS.EQ.'J') GOTO 5
0179          TYPE 115
0180      115 FORHAT(' DAN GEBRUIKEN WE HET OUDE PROEFNUMMER')
0181          PROEF=KONREC
0182          GOTO 5

```

```

C
0183      22 TYPE 111
0184      111 FORMAT(' WELK PROEFNUMMER IS GOED 1 OF 2 ? '$)
0185          ACCEPT *,IANS
0186          IF(IANS.EQ.2) GOTO 23
0187          PROEF=KONREC
0188          GOTO 3

```

```

C
0189      23 BACKSPACE 11
0190          READ(11,*) PRO,(NHULP(I),I=1,9)
0191          BACKSPACE 11
0192          WRITE(11,*) PROEF,(NHULP(I),I=1,9)
0193          NREC=1
0194          GOTO 3

```

```

C-----
C                                     EINDE SUBROUTINE SPLITS
C-----

```

```

0195      999 TYPE *, 'E-O-F GELEZEN IN FILE ',UNIT
0196          DO 550 I=1,NSPLIT
0197          J=I+10

```

FORTRAN IV-PLUS V3.0-3
LOGGER.FTN#12 /TR:BLOCKS/WR

10:33:58

7-Jan-82

0198 550 CLOSE(UNIT=J)
0199 RETURN
0200 END

BiJase 5.

SUBROUTINE

BETTER


```
0001      SUBROUTINE BETTER(M,UNIT,MREC1)
C
C-----
C
C      SUBROUTINE BETTER                                K.OOSTINDIE ICW
C
C      DEZE ROUTINE ZORGT VOOR HET VERBETEREN VAN DE FOUTE REGISTRATIES UIT EEN
C      REGISTRATIEFILE. EEN HELE SCAN WORDT IN DE SUBROUTINE MEEGENOMEN.
C      RECORD VOOR RECORD KAN NU BEKEKEN WORDEN EN BESLIST WORDEN WAT ERMEE
C      MOET GEBEUREN.
C
C-----
C
0002      INTEGER UNIT
0003      BYTE REEKS(80),IREEKS(80)
0004      REAL MEET
0005      COMMON MEET(100),MEETW(100),KAN(100)
C
0006      DO 60 I=1,MREC1
0007      60 BACKSPACE UNIT
0008      UNIT=19
0009      N=0
0010      REWIND 19
0011      TYPE 201
0012      201 FORMAT(' WAT WILT U MET HET NA-VOLGENDE RECORD DOEN?')
          *' 1 = RECORD BLIJFT ONGEWIJZIGD '//
          *' 2 = RECORD WORDT WEGGEGOOID '//
          *' 3 = VOORAFGAAND AAN DIT RECORD, 1 OF MEER RECORDS TUSSENVOEGEN(
          *AFSLUITEN MET )'//
          *' 4 = VOLGEND OP DIT RECORD, 1 OF MEER RECORDS TUSSENVOEGEN(AFBL
          *UITEN MET )'//
          *' 5 = BINNEN DIT RECORD WORDT ER VERBETERD '//
C
0013      1 READ(20,102,END=999) KTAL,(REEKS(I),I=1,KTAL)
0014      TYPE 101,(REEKS(I),I=1,KTAL)
0015      100 FORMAT(80A1)
0016      101 FORMAT(1H ,80A1)
0017      203 FORMAT(' WELK VAN DE 5 MOGELIJKHEDEN WENST U VOOR DIT RECORD TE GE
          *BRUIKEN ?')
C-----
C
C      KEUZE MOGELIJKHEDEN VOOR HET VERBETEREN VAN DE REGISTRATIES
C-----
0018      2 TYPE 203
0019      ACCEPT *,IG
0020      IF(IG.LT.1.OR.IG.GT.5) GOTO 2
0021      GOTO(3,4,5,6,7),IG
0022      3 WRITE(19,101)(REEKS(I),I=1,KTAL)
0023      N=N+1
0024      GOTO 51
C
0025      4 GOTO 51
C
0026      5 ACCEPT 102,JTAL,(IREEKS(I),I=1,JTAL)
0027      N=N+1
0028      IF(IREEKS(1).EQ.1H) GOTO 3
0029      WRITE(19,101)(IREEKS(I),I=1,JTAL)
```

```
0030      GOTO 5
      C
0031      6 WRITE(19,101)(REEKS(I),I=1,KTAL)
0032      N=N+1
0033      15 ACCEPT 102,JTAL,(IREEKS(I),I=1,JTAL)
0034      IF(IREEKS(1).EQ.1H) GOTO 51
0035      N=N+1
0036      WRITE(19,101)(IREEKS(I),I=1,JTAL)
0037      GOTO 15
      C
0038      7 TYPE 202
0039      202 FORMAT(' TIK IN OP DE MANIER VAN .....=..... '//
      *' (= IS VERPLICHT ALS SCHEIDINGSTEKEN)')
0040      ACCEPT 102,K,(IREEKS(I),I=1,K)
0041      102 FORMAT(Q,80A1)
0042      8 DO 10 I=1,K
0043      10 IF(IREEKS(I).EQ.1H=) GOTO 9
0044      9 NREEKS=I-2
0045      DO 30 I=1,80
0046      IF(REEKS(I).EQ.IREEKS(1)) GOTO 11
0047      GOTO 30
0048      11 IF(NREEKS.EQ.0) GOTO 12
0049      DO 20 J=1,NREEKS
0050      IF(REEKS(I+J).NE.IREEKS(J+1)) GOTO 30
0051      IF(J.EQ.NREEKS) GOTO 12
0052      20 CONTINUE
0053      30 CONTINUE
0054      12 NREEKS=NREEKS+3
0055      KREEKS=I-1
0056      IF(I+NREEKS-3.NE.KTAL) GOTO 13
0057      WRITE(19,101) (REEKS(I),I=1,KREEKS),(IREEKS(J),J=NREEKS,K)
0058      N=N+1
0059      GOTO 14
      C
0060      13 L1=-1+KREEKS+NREEKS
0061      IF(KREEKS.EQ.0) GOTO 16
0062      WRITE(19,101)(REEKS(I),I=1,KREEKS),(IREEKS(J),J=NREEKS,K),(REEKS(L
      *),L=L1,KTAL)
0063      N=N+1
0064      GOTO 14
      C
0065      16 WRITE(19,101)(IREEKS(J),J=NREEKS,K),(REEKS(L),L=L1,KTAL)
0066      14 DO 50 I=1,80
0067      REEKS(I)=1H
0068      IREEKS(I)=1H
0069      50 CONTINUE
0070      51 IF(M.GT.N) GOTO 1
0071      REWIND 19
0072      999 RETURN
0073      END
```

Bijlage 6.

SUBROUTINE

HULP

0001 SUBROUTINE HULP(IROUT)

```
C
C-----
C
C          SUBROUTINE HULP          K. OOSTINDIE  ICW
C
C  SUBROUTINE HULP ZORGT VOOR DE IN- EN UITVOER VAN EN NAAR DE SPLITSFILES
C  IN DEZE SUBROUTINE MOET DE KOPPELING GELEED WORDEN VOOR EEN NIEUWE
C  YKINGS EN TABELLERINGSSUBROUTINE
C-----
C
```

```
0002      BYTE NFILE(35),NFILE2(35),HELP(2)
0003      REAL MEET
0004      INTEGER PROEF
0005      COMMON MEET(100),MEETW(100),KAN(100)
0006      COMMON NWIS(20),JDAG(10),JUUR(10)
```

```
C
0007      JROUT=1
0008      N=1
0009      NTAL=1
0010      IF(IROUT.EQ.2.OR.IROUT.EQ.4) GOTO 26
```

```
C-----
C
C          KEUZE DATALOGGER
C-----
```

```
0011      3 TYPE 102
0012      102 FORMAT(' TIK IN VOOR DATALOGGER: '//
*           ' REMOTE SENSING EEN 1'//
*           ' BORSSELE          EEN 2'//)
0013      TYPE *, ' T '
0014      ACCEPT *,ILOG
0015      IF(ILOG.LT.1.OR.ILOG.GT.2) GOTO 3
0016      IF(ILOG.NE.2) GOTO 26
```

```
C
0017      TYPE 106
0018      106 FORMAT(' WILT U 1 WAARNEKING PER ETMAAL ? '$)
0019      ACCEPT 101,HELP(1)
0020      IF(HELP(1).EQ.1HJ) GOTO 27
```

```
C
0021      26 IF(IROUT.EQ.1) GOTO 7
0022      HELP(1)=1H
0023      HELP(2)=1H
0024      NMAX=0
0025      N1=0
0026      NSPLI=0
0027      IFST=10
0028      NDAG1=0
```

```
C-----
C
C          OPENEN SPLITSFILES
C-----
```

```
0029      1 TYPE 100
0030      100 FORMAT(' HOE HEET UW EERSTE SPLITS-FILE? '$)
0031      ACCEPT 101,NFILE
0032      NFILE(35)=0
0033      OPEN(UNIT=IFST,NAME=NFILE,TYPE='OLD',ERR=1)
0034      101      FORMAT(35A1)
```

```
0035 REWIND IFST
0036 6 READ(IFST,300) TEST,NKAN,(KAN(I),I=1,6)
0037 300 FORMAT(F11.2,3X,I2,13X,6I7)
0038 IF(TEST.LT.-99999.) GOTO 11
0039 CLOSE(UNIT=IFST,DISP='KEEP')
0040 GOTO 1
```

```
C
0041 11 DO 30 K=1,35
0042 IF(NFILE(K).EQ.1H) GOTO 17
0043 30 CONTINUE
0044 17 K=K+1
0045 IF(NFILE(K+1).NE.1H ) GOTO 18
0046 DECODE(1,104,NFILE(K)) NVER
0047 GOTO 19
0048 18 HELP(1)=NFILE(K)
0049 HELP(2)=NFILE(K+1)
0050 DECODE(2,105,HELP) NVER
0051 19 NSPLIT=NKAN-6
0052 NSPLTS=NSPLIT/10
0053 IF(NSPLTS*10.NE.NSPLIT) NSPLTS=NSPLTS+1
0054 IF(NMAX.LT.NSPLTS) NMAX=NSPLTS
0055 IF(NMAX-NSPLI.LE.0) GOTO 16
0056 NSPLI=NSPLI+1
0057 N1=N1+1
0058 DO 2 I=N1,NSPLTS
0059 J=I+1
0060 KFST=IFST+I
0061 NVER=NVER+1
0062 JVER=NVER
0063 DO 40 J=10,90,10
0064 IF(JVER.LT.10) GOTO 21
0065 40 JVER=JVER-J
0066 21 IF(JVER.EQ.8) NVER=NVER+2
0067 IF(NVER.GE.10) GOTO 22
0068 ENCODE(1,104,NFILE(K)) NVER
0069 GOTO 23
0070 22 ENCODE(2,105,HELP) NVER
0071 NFILE(K)=HELP(1)
0072 NFILE(K+1)=HELP(2)
0073 104 FORMAT(I1)
0074 105 FORMAT(I2)
0075 23 OPEN(UNIT=KFST,NAME=NFILE,TYPE='OLD')
0076 2 CONTINUE
```

C
C

KANALEN LEZEN

```
0077 16 IFST=10
0078 IF(NSPLTS.GE.N1) N1=NSPLTS
0079 NFST=7
0080 NLST=16
0081 IF(NLST.GT.NKAN) NLST=NKAN
0082 NSPLI=NSPLTS
0083 DO 10 I=1,NSPLTS
0084 IFST=IFST+1
0085 READ(IFST,*) (KAN(J),J=NFST,NLST)
0086 NFST=NLST+1
```

DEEL2.FTN)1

/TR:BLOCKS/WR

```

0087          NLST=NLST+10
0088          IF(NLST.GT.NKAN) NLST=NKAN
0089          CONTINUE
C-----
C
C                                     KLOK REGISTRATIE LEZEN
C-----
0090          10      DO 20 I=1,100
0091                    MEETW(I)=-32767
0092                    CONTINUE
0093                    NFST=7
0094                    NLST=16
0095                    IF(NLST.GT.NKAN) NLST=NKAN
0096                    IFST=10
0097                    READ(IFST,*,END=999) TEST,NDAG,NUUR,MIN,(MEETW(KAN(I)),I=1,6)
0098                    IF(TEST.LE.-99999.) GOTO 4
0099                    PROEF=TEST
0100                    GOTO 5
C-----
C
C                                     NIEUWE KANAALINDELING
C-----
0101          4      BACKSPACE IFST
0102                    JROUT=2
0103                    IF(NKAN.GE.5) BACKSPACE IFST
0104                    GOTO 6
C-----
C
C                                     MEETWAARDEN LEZEN
C-----
0105          5      DO 8 J=1,NSPLTS
0106                    IFST=IFST+1
0107                    READ(IFST,*)(MEETW(KAN(I)),I=NFST,NLST)
0108                    NFST=NLST+1
0109                    NLST=NLST+10
0110                    IF(NLST.GT.NKAN) NLST=NKAN
0111          8      CONTINUE
C-----
C
C                                     AANROEPEN VAN YKINGS EN TABELLERINGSSUBROUTINE'S
C-----
0112          IF(IROUT.EQ.1) RETURN
0113          IF(IROUT.EQ.2) GOTO 24
0114          IF(IROUT.EQ.4) GOTO 28
C-----
C
C                                     WELKE LOGGER?
C-----
0115          GOTO(60,61),ILOG
C-----
C
C                                     DATALOGGER REMOTE SENSING
C-----
0116          60      IF(IROUT.EQ.3) OPEN(UNIT=1,NAME='OUTPUT.TAB',TYPE='NEW')
0117                    CALL YK(NKAN,NSPLTS,PROEF,NDAG,NUUR,MIN,NDAG1)
0118                    IROUT=0
0119                    GOTO 7
C-----
C
C                                     DATALOGGER BORSSELE
C-----
0120          61      IF(IROUT.NE.3) GOTO 13
0121                    OPEN(UNIT=7,NAME='TABEL1.OUT',TYPE='NEW')
0122                    OPEN(UNIT=8,NAME='TABEL2.OUT',TYPE='NEW')
0123                    OPEN(UNIT=9,NAME='TABEL3.OUT',TYPE='NEW')

```

```

0124      OPEN(UNIT=6,NAME='TABEL4.OUT',TYPE='NEW')
0125      13 CALL YK2(NKAN,NSPLTS,PROEF,NDAG,NUUR,MIN,NDAG1,NTAL,PRINT)
0126      IROUT=0
0127      N=N+1
0128      NTAL=NTAL+1
0129      IF(N.NE.55) GOTO 14
0130      N=1
0131      NDAG1=0
0132      GOTO 7
-----
C
C                                     HIERVOOR NIEUWE LOGGERS AANKOPPELEN
C
-----
0133      14 GOTO 7
-----
C
C                                     AANROEPEN SUBROUTINE WISSEL
C
-----
0134      24 CALL WISSEL(JROUT,NDAG,NUUR)
0135      GOTO 29
-----
C
C                                     AANROEPEN SUBROUTINE INTVAL
C
-----
0136      28      CALL INTVAL(JROUT,NKAN)
-----
C
C                                     NIEUWE SPLITSFILES MAKEN
C
-----
0137      29 IF(JROUT.EQ.0) GOTO 31
0138      IF(JROUT.EQ.2) GOTO 32
0139      TYPE 107
0140      107 FORMAT(' HOE HETEN DE NIEUWE SPLITS-FILES ? ')
0141      ACCEPT 101,NFILE2
0142      NFILE2(35)=0
0143      N2=1
0144      32 N3=NSPLTS+1
0145      IF(N2-N3.GT.0) GOTO 33
0146      DO 35 I=N2,N3
0147      J=I+14
0148      35 OPEN(UNIT=J,NAME=NFILE2,TYPE='NEW')
-----
C
C                                     KANALEN MET MEETWAARDEN WEGSCHRIJVEN NAAR DE NIEUWE SPLIBFILES
C
-----
0149      33 NFST=7
0150      NLST=16
0151      TEST=-99999.99
0152      WRITE(15,300) TEST,NKAN,(KAN(K),K=1,6)
0153      DO 45 I=2,N3
0154      J=I+14
0155      WRITE(J,*) (KAN(K),K=NFST,NLST)
0156      NFST=NLST+1
0157      NLST=NLST+10
0158      IF(NLST.GT.NKAN) NLST=NKAN
0159      45 CONTINUE
0160      31 NFST=7
0161      NLST=16
0162      WRITE(15,*) PROEF,NDAG,NUUR,MIN,(MEETW(KAN(I)),I=1,6)
0163      DO 55 I=2,N3
0164      J=I+14

```

DEEL2.FTN:1

/TR:BLOCKS/WR

```

0165      WRITE(J,*) (MEETW(KAN(K)),K=NFST,NLST)
0166      NFST=NLST+1
0167      NLST=NLST+10
0168      IF(NLST.GT.NKAN) NLST=NKAN
0169      55 CONTINUE
0170      N2=N3+1
0171      JROUT=0
0172      GOTO 7

```

C-----

C

EINDE HULP

```

0173      999      NSPLTS=NSPLTS+1
0174      IFST=9
0175      ILST=14
0176      DO 9 I=1,NSPLTS
0177      IFST=IFST+1
0178      IF(I.NE.1) GOTO 36
0179      TYPE 108
0180      ACCEPT 101,HELP(1)
0181      36      IF(HELP(1).EQ.1HJ) GOTO 34
0182      CLOSE(UNIT=IFST,DISP='DELETE')
0183      GOTO 37
0184      34      CLOSE(UNIT=IFST,DISP='KEEP')
0185      37      IF(IROUT.EQ.3) GOTO 9
0186      ILST=ILST+1
0187      108     FORMAT(' WILT U DE OUDE SPLITS-FILES BEWAREN ?')
0188      CLOSE(UNIT=ILST,DISP='KEEP')
0189      9      CONTINUE
0190      IF(ILOG.NE.2) GOTO 15
0191      DO 25 I=6,9
0192      25     CLOSE(UNIT=I,DISP='SAVE')
0193      TYPE 106
0194      ACCEPT 101,HELP(1)
0195      IF(HELP(1).NE.1HJ) GOTO 15
0196      27     CALL ACHT
0197      15     IROUT=100
0198      RETURN
0199      END

```


Bijlage 7.

SUBROUTINE

WISSEL

DEEL3.FTN)1

/TR:BLOCKS/WR

```

0001      SUBROUTINE WISSEL(JROUT,NDAG,NUUR)
C
C-----
C
C          SUBROUTINE WISSEL                      K.OOSTINDIE      ICW
C
C          SUBROUTINE WISSEL ZORGT ERVOOR DAT TWEE OF MEER KANALEN ONDERLING MET
C          ELKAAR KUNNEN WORDEN VERWISSELD
C          HET IS MOGELIJK VANAF EEN OP TE GEVEN TIJDSTIP TE VERWISSELEN
C-----
C
0002      REAL MEET
0003      COMMON MEET(100),MEETW(100),KAN(100)
0004      COMMON NWIS(20),JDAG(10),JUUR(10)
C
0005      IF(JROUT.EQ.0) GOTO 1
C-----
C          OPVRAGEN VAN DE KANALEN DIE VERWISSELD MOETEN WORDEN MET HET TIJDSTIP
C-----
0006      TYPE 103
0007      103 FORMAT(' GEEF NU DE KANALEN OP DIE VERWISSELD MOETEN WORDEN''
* AFSLUITEN MET 9999'' WANNEER VAN AF HET BEGIN MOET WORDEN
* VERWISSELD VOOR DAG EEN 0 INTIKKEN')
0008      N=0
0009      DO 30 I=1,20,2
0010      TYPE 104
0011      ACCEPT *,NWIS(I)
0012      IF(NWIS(I).EQ.9999) GOTO 1
0013      TYPE 105
0014      ACCEPT *,NWIS(I+1)
0015      TYPE 106
0016      N=N+1
0017      ACCEPT *,JDAG(N)
0018      IF(JDAG(N).EQ.0) GOTO 30
0019      TYPE 107
0020      ACCEPT *,JUUR(N)
0021      30 CONTINUE
0022      104 FORMAT(1H , 'KANAALNR :')
0023      105 FORMAT(1H 'VERWISSELEN MET KANAALNR : ')
0024      106 FORMAT(1H , 'BEGINDAG :')
0025      107 FORMAT(1H , 'BEGINUUR :')
C-----
C          HET VERWISSELEN VAN DE KANALEN VANAF EEN BEPAALD TIJDSTIP
C-----
0026      1 IF(N.EQ.0) RETURN
0027      DO 10 I=1,2*N,2
0028      IF(JDAG((I+1)/2).NE.0) GOTO 2
0029      3 HULP=MEETW(NWIS(I))
0030      MEETW(NWIS(I))=MEETW(NWIS(I+1))
0031      MEETW(NWIS(I+1))=HULP
0032      GOTO 10
0033      2 IF(JDAG((I+1)/2).GT.NDAG) GOTO 10
0034      IF(JDAG((I+1)/2).NE.NDAG) GOTO 3
0035      IF(JUUR((I+1)/2).LE.NUUR) GOTO 3
0036      10 CONTINUE

```

FORTRAN IV-PLUS V3.0-3

10133134

12-Jan-82

DEEL3.FTN11

/TR:BLOCKS/WR

0037

RETURN

0038

END

BiJase 8.

SUBROUTINE

INTVAL

```

0001          SUBROUTINE INTVAL(JROUT,NKAN)
C
C-----
C
C          SUBROUTINE INTVAL          K.OOSTINDIE  ICW
C
C SUBROUTINE INTVAL ZORGT VOOR HET TESTEN OP INTERVALLEN. WANNEER EEN MEET-
C WAARDE IN EEN BEPAALD INTERVAL LIGT KAN DEZE WAARDE VERVANGEN DOOR EEN
C OP TE GEVEN CONSTATE.
C
C-----
C
0002          BYTE NFILE(35)
0003          DIMENSION IOND(20),IBOV(20),NEW(20),NR(20)
0004          INTEGER TI
0005          REAL MEET
0006          COMMON MEET(100),MEETW(100),KAN(100)
C
0007          IF(JROUT.NE.1) GOTO 2
C-----
C          OPVRAGEN VAN BEGEVENS VOOR HET WISSELEN VAN KANALEN
C-----
0008          N=0
0009          TI=0
0010          101  FORMAT(' SUBROUTINE INTVAL VOOR HET TESTEN OP INTERVAL WAARDEN'//
* DE INPUT VOOR DEZE SUBROUTINE DIENST ALS VOLGT TE WORDEN OPGEVEVE
* N ://
* KANAALNUMMER ONDERGRENDS BOVENGRENDS VERVANGENDE-WAARDE//
* WANNEER VOOR ALLE KANALEN OP EEN ZELFDE INTERVAL GETEST//
* MOET WORDEN , DIENST U IN PLAATS VAN HET KANAALNR. HET GETAL 9999
* IN TE VOEREN.//
* WANNEER U IN PLAATS VAN HET KANAALNR EEN NEGATIEF GETAL MET EEN
*/// INVOERT BETEKENT DIT EINDE INPUT'
*///)
0011          102  FORMAT(' VANAF WELKE FILE WORDEN DE INPUTGEGEVENS GELEZEN?'//
* INGEVAL U VANAF DE TERMINAL DE GEGEVENS IN WILT LEZEN'//
* MOET U VOOR FILE-NAAM : TI: INVULLEN'// FILE-NAAM =')
0012          TYPE 101
0013          TYPE 102
0014          ACCEPT 100,NFILE
0015          100  FORHAT(35A1)
0016          NFILE(35)=0
0017          OPEN(UNIT=19,NAME=NFILE,TYPE='OLD')
0018          IF(NFILE(1).EQ.1HT.AND.NFILE(2).EQ.1HI.AND.NFILE(3).EQ.1H;) TI=1
0019          DO 10 I=1,20
0020          IF(TI.EQ.0) GOTO 1
0021          TYPE 103
0022          1    READ(19,*) NR(I),IOND(I),IBOV(I),NEW(I)
0023          IF(NR(I).LT.0) GOTO 2
0024          N=N+1
0025          10   CONTINUE
0026          103  FORMAT(' TIK IN: KANAALNR. ONDER- EN BOVENGRENDS EN VERVANGENDE W
*WAARDE'//
* OF : EEN NEGATIEF GETAL MET EEN / VOOR EINDE INVOER'//
* INVOER =')
C-----

```

C HET WISSELEN VAN DE KANALEN
C-----

```
0027      2      IF(N.EQ.0) RETURN
0028          DO 20 I=1,N
0029          IF(NR(I).EQ.9999) GOTO 3
0030          IF(MEETW(NR(I)).GE.IOND(I).AND.
          * MEETW(NR(I)).LE.IBOV(I)) MEETW(NR(I))=NEW(I)
0031          GOTO 20
0032      3      DO 30 J=1,NKAN
0033          IF(MEETW(KAN(J)).GE.IOND(I).AND.
          * MEETW(KAN(J)).LE.IBOV(I)) MEETW(KAN(J))=NEW(I)
0034          30      CONTINUE
0035          20      CONTINUE
0036          RETURN
0037          END
```

Bijlage 9.

SUBROUTINE

YK2

```

0001      SUBROUTINE YK2(NKAN,NSPLTS,PROEF,NDAG,NUUR,MIN,NDAG1,NTAL,PRINT)
C
C-----
C
C          SUBROUTINE YK2                      K.OOSTINDIE      ICW
C
C  SUBROUTINE YK2 ZORGT ERVOOR DAT DE MEETWAARDEN VAN DATALOGGER BORSSELE
C  OMGEZET WORDEN VAN MILIVOLTS NAAR DE GEWENSTE GROOTHEDEN EN DAARNA
C  OP OVERZICHTELIJKE WIJZE BETABELLEERD WORDEN
C
C-----
C
0002      IMPLICIT INTEGER(A-H,O-Z)
0003      BYTE DAT(9)
0004      REAL MEET,REEKS(100)
0005      COMMON MEET(100),MEETW(100),KAN(100)
C-----
C          OMZETTEN VAN DE MEETWAARDEN IN MILIVOLTS NAAR DE GEWENSTE GROOTHEDEN
C-----
0006      CALL DATE(DAT)
0007      DO 10 I=1,40
0008          IF(MEETW(I).LT.0) MEETW(I)=-32767
0009      10 CONTINUE
0010      DO 15 I=1,40
0011          IF(I.EQ.1.OR.(I.GE.4.AND.I.LT.8).OR.I.EQ.13.OR.I.EQ.9) GOTO 15
0012          IF(MEETW(I).GE.1520) MEETW(I)=-32767
0013          IF(MEETW(I).GT.1500) MEETW(I)=1500
0014      15 CONTINUE
C
0015      REEKS(43)=4H *
0016      REEKS(44)=4H *
0017      IF(MEETW(8).EQ.-32767) GOTO 41
0018      DRAIN1=14.715*EXP(.0005*MEETW(8))+1
0019      ENCODE(4,104,REEKS(44)) DRAIN1
0020      DRAINM=DRAIN1*22.3
0021      ENCODE(4,104,REEKS(45)) DRAINM
0022      41 REEKS(1)=4H *
0023          IF(MEETW(1).EQ.-32767) GOTO 1
0024          MEET1=10*MEETW(1)
0025          ENCODE(4,104,REEKS(1)) MEET1
0026      1 REEKS(2)=4H *
0027          IF(MEETW(2).EQ.-32767) GOTO 2
0028          MEETW(2)=.1*MEETW(2)-62
0029          MEET2=119-MEETW(2)
0030          ENCODE(4,104,REEKS(2)) MEET2
0031      2 REEKS(3)=4H *
0032          IF(MEETW(3).EQ.-32767) GOTO 3
0033          MEETW(3)=.1*MEETW(3)-56
0034          MEET3=139-MEETW(3)
0035          ENCODE(4,104,REEKS(3)) MEET3
0036      3 REEKS(10)=4H *
0037          IF(MEETW(10).EQ.-32767) GOTO 6
0038          IF(NDAG.GE.341) GOTO 4
0039          IF(NDAG.GT.340.AND.NUUR.GE.14) GOTO 4
0040          MEETW(10)=.1*MEETW(10)-55
0041      GOTO 5

```



```
0042      4 MEETW(10)=.1*MEETW(10)-25
0043      5 MEET10=125-MEETW(10)
0044      ENCODE(4,104,REEKS(10)) MEET10
0045      6 REEKS(11)=4H *
0046      IF(MEETW(11).EQ.-32767) GOTO 7
0047      MEETW(11)=.1*MEETW(11)-69
0048      MEET11=119-MEETW(11)
0049      ENCODE(4,104,REEKS(11)) MEET11
0050      7 REEKS(12)=4H *
0051      IF(MEETW(12).EQ.-32767) GOTO 11
0052      IF(NDAG.GE.341) GOTO 8
0053      IF(NDAG.EQ.340.AND.NUUR.GE.14) GOTO 8
0054      MEETW(12)=.1*MEETW(12)-65
0055      GOTO 9
0056      8 MEETW(12)=.1*MEETW(12)-35
0057      9 MEET12=125-MEETW(12)
0058      ENCODE(4,104,REEKS(12)) MEET12
0059      11 MEETW(13)=4H *
0060      DO 20 I=4,6
0061      20 MEETW(I)=-32767
0062      MEETW(8)=-32767
0063      MEETW(9)=-32767
0064      REEKS(14)=4H *
0065      IF(MEETW(14).EQ.-32767) GOTO 14
0066      IF(NDAG.GE.341) GOTO 12
0067      IF(NDAG.EQ.340.AND.NUUR.GE.14) GOTO 12
0068      MEETW(14)=.1*MEETW(14)-80
0069      GOTO 13
0070      12 MEETW(14)=.1*MEETW(14)-50
0071      13 MEET14=118-MEETW(14)
0072      ENCODE(4,104,REEKS(14)) MEET14
0073      14 REEKS(15)=4H *
0074      IF(MEETW(15).EQ.-32767) GOTO 16
0075      MEETW(15)=.1*MEETW(15)-65
0076      MEET15=124-MEETW(15)
0077      ENCODE(4,104,REEKS(15)) MEET15
0078      16 REEKS(16)=4H *
0079      IF(MEETW(16).EQ.-32767) GOTO 17
0080      MEETW(16)=.1*MEETW(16)-119
0081      MEET16=120-MEETW(16)
0082      ENCODE(4,104,REEKS(16)) MEET16
0083      17 REEKS(21)=4H *
0084      IF(MEETW(21).EQ.-32767) GOTO 18
0085      MEETW(21)=.3*MEETW(21)-133
0086      MEET21=70-MEETW(21)
0087      ENCODE(4,104,REEKS(21)) MEET21
0088      18 REEKS(22)=4H *
0089      IF(MEETW(22).EQ.-32767) GOTO 19
0090      MEETW(22)=.29*MEETW(22)-132
0091      MEET22=75-MEETW(22)
0092      ENCODE(4,104,REEKS(22)) MEET22
0093      19 REEKS(23)=4H *
0094      IF(MEETW(23).EQ.-32767) GOTO 21
0095      MEETW(23)=.3*MEETW(23)-133
0096      MEET23=72-MEETW(23)
0097      ENCODE(4,104,REEKS(23)) MEET23
```

BORSYK.FTN12

/TR:BLOCKS/WR

```

0098      21 REEKS(24)=4H *
0099      IF(MEETW(24).EQ.-32767) GOTO 22
0100      MEETW(24)=.3*MEETW(24)-113
0101      MEET24=72-MEETW(24)
0102      ENCODE(4,104,REEKS(24)) MEET24
0103      22 REEKS(31)=4H *
0104      IF(MEETW(31).EQ.-32767) GOTO 23
0105      MEETW(31)=.3*MEETW(31)-116
0106      MEET31=110-MEETW(31)
0107      ENCODE(4,104,REEKS(31)) MEET31
0108      23 REEKS(32)=4H *
0109      IF(MEETW(32).EQ.-32767) GOTO 24
0110      MEETW(32)=.29*MEETW(32)-114
0111      MEET32=115-MEETW(32)
0112      ENCODE(4,104,REEKS(32)) MEET32
0113      24 REEKS(33)=4H *
0114      IF(MEETW(33).EQ.-32767) GOTO 26
0115      MEETW(33)=.31*MEETW(33)-108
0116      MEET33=113-MEETW(33)
0117      ENCODE(4,104,REEKS(33)) MEET33
0118      26 CONTINUE

```

C

C

REGENMETER

```

0119      IF(PRINT.EQ.0) GOTO 27
0120      AAND=1H
0121      DAYMM=0
0122      27 SCAN=0
0123      ACHT=4H *
0124      PRINT=0
0125      IF(MEETW(7).EQ.-32767) GOTO 28
0126      METOLD=METER
0127      METER=MEETW(7)/20.
0128      ENCODE(4,104,REEKS(40)) METER
0129      IF(NTAL.EQ.1) METOLD=METER
0130      29 IF(
      *( NUUR.EQ.7.AND.MIN.BT.50).OR.
      *( NUUR.EQ.8.AND.MIN.LT.10)) PRINT=1
0131      IF(AAND.EQ.1H?) GOTO 31
0132      SCAN=METER-METOLD
0133      IF(SCAN.LT.-1) GOTO 32
0134      33 DAYMM=DAYMM+SCAN
0135      ENCODE(4,104,REEKS(41)) DAYMM
0136      IF(SCAN.EQ.1) SCAN=0
0137      GOTO 39
0138      28 AAND=1H?
0139      GOTO 29
0140      31 REEKS(40)=4H *
0141      REEKS(41)= 4H *
0142      GOTO 39
0143      32 KIEP=1
0144      SCAN=100+METER-METOLD
0145      GOTO 33

```

C

C

UITPRINTEN VAN DE TABELLEN

```

0144 100 FORMAT(1H1,' REGISTRATIEVERWERKING'70X'DATALOGGER BORSSELE '9A1)
0147 101 FORMAT(1H 'MEET-OPSTELLING BORSSELE NEERSLAG AF
*VOERSNELHEID GELEIDINGSVERMOGEN VOCHT RECORD
*NUMMER' // EERSTE TABEL -----
*
* IN DE'// METER /SCAN /DAG DRAIN 1
* 2 1 2 DAG UUR MIN HG IN T
*EKEN = DATA'// DAG UUR MIN 0.1MM 0.1MM 0.1MM
* ML/S ML/S MUM/D MUM/D MV CL/L MV UREN OF
*RECORDS -FILE'// |-----|-----|-----|-----|
*|-----|-----|-----|-----|-----|-----|-----|
*|-----|-----|-----|-----|-----|-----|-----|
*RSPRINGEN -----//)
0148 200 FORMAT(1H 'MEET-OPSTELLING BORSSELE WATERSTANDEN TOV.
*NAP ZWEEMER WATERSTANDEN TOV. NAP GOENSE
*NUMMER'// ' TWEDE TABEL -----
*
* IN DE'// SLOOT GRWST 1A 2A 3A
* 4A 5A 9A 13A SLOOT GRWST 21 22 23
* DATA-'// ' DAG UUR MIN CH CH CH
* CM CM CM CM CM CH CM CM CM CM
* -FILE'// |-----|-----|-----|-----|-----|-----|
*|-----|-----|-----|-----|-----|-----|-----|
*|-----|-----|-----|-----|-----|-----|-----|
*|-----//)
0149 300 FORMAT(1H 'MEETOPSTELLING BORSSELE WATERSTANDEN MIN MAAIVE
*LD ZWEEMER WATERSTANDEN MIN MV. GOENSE
* NUMMER'// ' DERDE TABEL -----
*
* IN DE'// SLOOT GRWST 1A 2A 3A
* 4A 5A 9A 13A SLOOT GRWST 21 22 23
* DATA-'// ' DAG UUR MIN CH CH CH
* CM CM CM CM CM CH CM CM CM CM
* -FILE'// |-----|-----|-----|-----|-----|-----|
*|-----|-----|-----|-----|-----|-----|-----|
*|-----|-----|-----|-----|-----|-----|-----|
*|-----//)
0150 400 FORMAT(1H 'MEETOPSTELLING BORSSELE WATERSTANDEN MIN MV. L
*IERE WATERSTANDEN TOV NAP NUMMER'// ' VIERDE TABEL
*
* IN DE'// GRWST 31 32 SLOOT DATA-'// ' DAG UUR MIN
* GRWST 31 32 SLOOT DATA-'// ' DAG UUR MIN
* CM CM CM CH CM CM CM CM -F
*FILE'// |-----|-----|-----|-----|-----|-----|
*|-----|-----|-----|-----|-----|-----|-----|
*|-----|-----|-----|-----|-----|-----|-----|
*|-----//)
0151 39 DO 70 I=1,100
0152 IF(MEETW(I).EQ.-32767) GOTO 79
0153 ENCODE(4,104,MEET(I)) MEETW(I)
0154 GOTO 70
0155 79 MEET(I)=4H *
0156 70 CONTINUE
0157 ENCODE(4,104,REEKS(43)) DAYMH
0158 REEKS(42)=4H *
0159 IF(NDAG1.EQ.0) GOTO 34
0160 35 IF(PRINT.EQ.1) GOTO 36
0161 WRITE(7,102) NDAG,NUUR,MIN,(REEKS(I),I=40,42),(REEKS(J),J=44,45)
*,MEET(1),REEKS(1),NTAL
0162 GOTO 37
0163 36 WRITE(7,103) NDAG,NUUR,MIN,(REEKS(I),I=40,41),REEKS(43),AAND,

```

```
      *(REEKS(J),J=44,45),MEET(1),REEKS(1),NTAL
0164 102 FORMAT(1H ,3(I5,1X),7X,3(A4,2X),6X,2(A4,8X),7X,2(A4,2X),30X,I6)
0165 103 FORMAT(1H ,3(I5,1X),7X,2(A4,2X),1X,A4,A1,6X,2(A4,8X),7X,2(A4,2X)
      *,30X,I6)
0166 104 FORHAT(I4)
0167 201 FORMAT(1H ,3(I5,1X),7X1A4,9X,7(A4,2X),5X,A4,8X,3(A4,2X),I5)
0168 401 FORMAT(1H ,3(I5,1X),13X,3(A4,2X),18X,3(A4,2X),5X,I5)
0169 37 WRITE(8,201) NDAG,NUUR,MIN,MEET(16),MEET(3),MEET(12),MEET(11)
      *,MEET(10),MEET(2),MEET(14),MEET(15),MEET(24),MEET(21),MEET(
      *22),MEET(23),NTAL
0170 38 WRITE(9,201) NDAG,NUUR,MIN,REEKS(16),REEKS(3),REEKS(12),REEKS(11)
      *,REEKS(10),REEKS(2),REEKS(14),REEKS(15),REEKS(24),REEKS(21)
      *,REEKS(22),REEKS(23),NTAL
0171 WRITE(6,401) NDAG,NUUR,MIN,REEKS(31),REEKS(32),REEKS(33),MEET(31
      *),MEET(32),MEET(33),NTAL
0172 NDAG1=NDAG
0173 RETURN
0174 34 WRITE(7,100) DAT
0175 WRITE(8,100) DAT
0176 WRITE(9,100) DAT
0177 WRITE(6,100) DAT
0178 WRITE(7,101)
0179 WRITE(8,200)
0180 WRITE(9,300)
0181 WRITE(6,400)
0182 GOTO 35
0183 END
```