

Technische Hogeschool Delft

Afdeling der Luchtvaart- en
Ruimtevaarttechniek
Delft

Prins Maurits Laboratorium

Organisatie voor Toegepast
Natuurwetenschappelijk Onderzoek
Rijswijk

Rapport LR-430

Rapport PML 1984-C57

SFCC publikatie nr. 14

Computerprogramma voor het
regel/bedieningssysteem van de
gastoevoerinstallatie voor de vaste brandstof
verbrandingskamer

R.J. van der Wal
P.A.O.G. Korting
H.F.R. Schöyer

Delft/Rijswijk

Juli 1984

Technische Hogeschool Delft
Afdeling der Luchtvaart- en
Ruimtevaarttechniek
Delft

Prins Maurits Laboratorium
Organisatie voor Toegepast
Natuurwetenschappelijk Onderzoek
Rijswijk

Rapport LR-430
Rapport PML 1984-C57
SFCC publikatie nr. 14

Computerprogramma voor het regel/bedieningssysteem van de gastoevoerinstallatie voor de vaste brandstof verbrandingskamer

R.J. van der Wal
P.A.O.G. Korting
H.F.R. Schöyer

Delft/Rijswijk

Juli 1984

SAMENVATTING

Voor het onderzoek naar het verbrandingsproces in een Vaste Brandstof Verbrandingskamer staat een gastoevoersysteem met een bijbehorend regel/bedienings-systeem ter beschikking.

Dit regel/bedieningssysteem is uitgerust met een microprocessor dat de opeenvolging van gebeurtenissen tijdens een experiment regelt.

Daarnaast vereenvoudigt deze microprocessor het in gereedheid brengen van het gastoevoersysteem ten behoeve van een experiment.

In dit rapport wordt de programmatuur voor de microprocessor toegelicht. Tevens zijn in dit rapport de programma listings opgenomen.

Dit rapport vormt een onderdeel van het project 'Investigation of a Solid Fuel Combustion Chamber'. Dit project wordt gefinancierd door de Stichting voor de Technische Wetenschappen (projektnr. DLR 14.0120), het projektbureau Energieonderzoek (projektnr. 90753.140), de Afdeling der Luchtvaart- en Ruimtevaarttechniek van de Technische Hogeschool Delft en het Prins Maurits Laboratorium TNO te Rijswijk.

INHOUDSOPGAVE

Samenvatting

1. Inleiding
2. De programmatuur
 - 2.1. Flowschema van het SFCC-programma
 - 2.2. Common blocks in het SFCC-programma
 - 2.3. Invoer van gegevens
 - 2.4. Verwerking van de gegevens
 - 2.5. Berekening van de massastromen
 - 2.6. Verrijgingspercentage
 - 2.7. Instelling van de drukken en massastromen
 - 2.8. Uitvoering van de proef
 - 2.9. Afwikkeling van de proef
 - 2.10. Het werken met interrupts
 - 2.11. Controle terugmelding elektrische kleppen
3. Foutmeldingen
4. Overige programmatuur
 - 4.1. Routines voor i/o- en a/d-, d/a-handelingen
5. Beschikbaarheid van de programma's
6. Referenties

1. INLEIDING

Voor het onderzoek naar het verbrandingsproces in een Vaste Brandstof Verbrandingskamer wordt gebruik gemaakt van een gastoevoersysteem. Dit gastoevoersysteem kan de volgende gassen aan de VBVK leveren (zie Figuur 1.1):

1. lucht;
2. zuurstof voor de verwarming van lucht in de vitiator met behulp van methaan;
3. zuurstof voor de ontsteking van de vitiator met behulp van methaan;
4. zuurstof voor de ontsteking van de VBVK met behulp van waterstof;
5. waterstof;
6. methaan voor de verwarming van lucht in de vitiator;
7. methaan voor de ontsteking van de vitiator.

Daarnaast wordt stikstof gebruikt voor het spoelen van de methaan en waterstoflijnen alsmede voor het sturen van pneumatische kleppen en reduceerventielen.

Een gedetailleerde beschrijving van dit gastoevoersysteem is te vinden in Ref. 1. Het regel/bedieningssysteem van dit gastoevoersysteem, zie Ref. 2, is uitgerust met een microprocessor (Figuur 1.2) die de opeenvolging van gebeurtenissen voor, tijdens en na een experiment met de VBVK regelt. Deze microprocessor stuurt niet alleen de elektro-pneumatische kleppen maar kent ook de druk (en soms de temperatuur) in de gasleidingen. Daarnaast zijn alle kleppen (ook handbediende) uitgerust met microswitches zodat de microprocessor de stand van alle kleppen kent.

Totaal is de microprocessor uitgevoerd met:

- 64 digitale ingangen (64 bits, genummerd 1 t/m 64) waarmee o.a. de klepstanden (door middel van microswitches) kunnen worden gecontroleerd;
- 64 digitale uitgangen (64 bits, genummerd 65 t/m 128) voor de aansturing van de elektrische kleppen, signalering, etc.;
- 64 analoge ingangen, waardoor de computer informatie krijgt over heersende drukken en temperaturen in het systeem;
- 4 analoge uitgangen, welke gebruikt worden voor de uitlezing van door de computer berekende massastromen.

In Tabellen 1.1 t/m 1.4 staat aangegeven met welke kleppen, microswitches, thermokoppels, drukopnemers e.d. deze in- en uitgangen verbonden zijn.

De microprocessor is zo geprogrammeerd dat deze een aantal verschillende soorten proeven kan uitvoeren, zie Tabel 1.5.

Het uitvoeren van deze proeven is voor de gebruiker betrekkelijk eenvoudig; naast het opgeven van de soort proef en enkele gegevens zoals brandtijd en gewenste oxydator massastroom gaat de microprocessor niet alleen na of zo'n proef uitgevoerd kan worden d.w.z. of bijv. alle (ook de handbediende) kleppen in de juiste stand staan, maar ook of er voldoende gas aanwezig is en of er voldoende druk is, etc. Mocht dat niet het geval zijn dan geeft de microprocessor de gebruiker instructies die er uiteindelijk toe moeten leiden dat een experiment kan worden uitgevoerd.

Tijdens een experiment controleert de microprocessor de druk en temperatuur op enkele vitale punten in het gastoevoersysteem. Mocht een druk of een temperatuur niet meer aan bepaalde criteria voldoen dan wordt het experiment voortijdig beëindigd.

Tabel 1.1. Digitale ingangen van de microprocessor.

bitnr.	kraan nr. 1)	klep nr. 1)	connectie
1	1		lucht
2	2		O ₂ 1e lijn
3	3		O ₂ 1e lijn
4	4		O ₂ 1e lijn
5	5		O ₂ 1e lijn
6	6		O ₂ 2e lijn
7	7		O ₂ 2e lijn
8	8		O ₂ 2e lijn
9	9		O ₂ 2e lijn
10	10		O ₂ 2e lijn
11	11		O ₂ 3e lijn
12	12		H ₂
13	13		H ₂
14	14		H ₂
15	15		H ₂
16	16		H ₂
17	17		H ₂
18	18		CH ₄ 1e lijn
19	19		CH ₄ 1e lijn
20	20		CH ₄ 1e lijn
21	21		CH ₄ 1e lijn
22	22		CH ₄ 1e lijn
23	23		lucht brandweer
24	24		lucht buffervaten
25		1	lucht
26		2	afblaasklep
27		3	waterkoeling
28		4	O ₂ 1e lijn
29		5	O ₂ 2e lijn
30		6	O ₂ 3e lijn
31		7	H ₂
32		8	H ₂
33		9	CH ₄ 1e lijn
34		10	CH ₄ 1e lijn
35		11	CH ₄ 2e lijn
36		12	CH ₄ 2e lijn
37		13	N ₂ (spoelen CH ₄ 1e lijn)
38		14	N ₂ (spoelen CH ₄ 2e lijn)
39		15	N ₂ (spoelen H ₂ lijn)
40			réserve
41			stand-by DA-systeem
42			voedingsspanning aanwezig
43			beveiligingsunit aan
44			
t/m			reserve
56			
57			start VBVK interrupt
58			stop VBVK interrupt
59			
t/m			reserve interrupts
64			

1) De kraan/klep nummers komen overeen met de nummers in Fig. 1.1.

Tabel 1.2. Digitale uitgangen van de microprocessor.

bitnr.	connectie	bitnr.	connectie
65	ready for air supply	102	6. 3e O ₂ (52/53)
66	ready for oxygen supply	103	7 H ₂ (72/73)
67	ready for hydrogen supply	104	8 H ₂ (80)
68	ready for methane supply	105	9 1e CH ₄ (101/102)
69	test phase	106	10 1e CH ₄ (110/111)
70	preparation phase	107	11 2e CH ₄ (113/114)
71	ignition vitiator	108	12 2e CH ₄ (120/121)
72	hot air supply	109	13 spoelen 2e CH ₄ (137)
73	ignition SFCC	110	14 spoelen 1e CH ₄ (135)
74	SFCC run	111	15 spoelen H ₂ (133)
75	purge	112	reserve
76		113	
t/m	reseve	t/m	reserve
80		117	
81		118	bougie vitiator
t/m	klepregeling	119	bougie VBVK
92		120	sirene
93	gereserveerd voor reset	121	stand by gassupply system
t/m	interrupt flags	122	stand by Data Acq. system
96		123	activate Data Acq. system
97	1 lucht (13/14)	124	reserve
98	2 afblaasklep	125	emergency stop
99	3 waterkoeling	126	check signal
100	4 1e O ₂ (33/34)	127	pilotflame on
101	5 2e O ₂ (42)	128	reserve

Tabel 1.3. Analoge ingangen van de microprocessor.

kanaal	connectie
1	primaire druk lucht
2	secundaire druk lucht
3	temperatuur lucht-SMRD
4	druk vitiator
5	temperatuur vitiator
6	temperatuur pilot vlam
7	primaire druk 1e O ₂ lijn
8	secundaire druk 1e O ₂ lijn
9	temperatuur O ₂ -SMRD
10	primaire druk 2e O ₂ lijn
11	secundaire druk 2e O ₂ lijn
12	secundaire druk 3e O ₂ lijn
13	primaire druk H ₂
14	secundaire druk H ₂
15	primaire druk 1e CH ₄ lijn
16	secundaire druk 1e CH ₄ lijn
17	temperatuur CH ₄ -SMRD
18	primaire druk 2e CH ₄ lijn
19	secundaire druk 2e CH ₄ lijn
20	druk VBVK
21	temperatuur VBVK
22	
t/m	reserve
64	

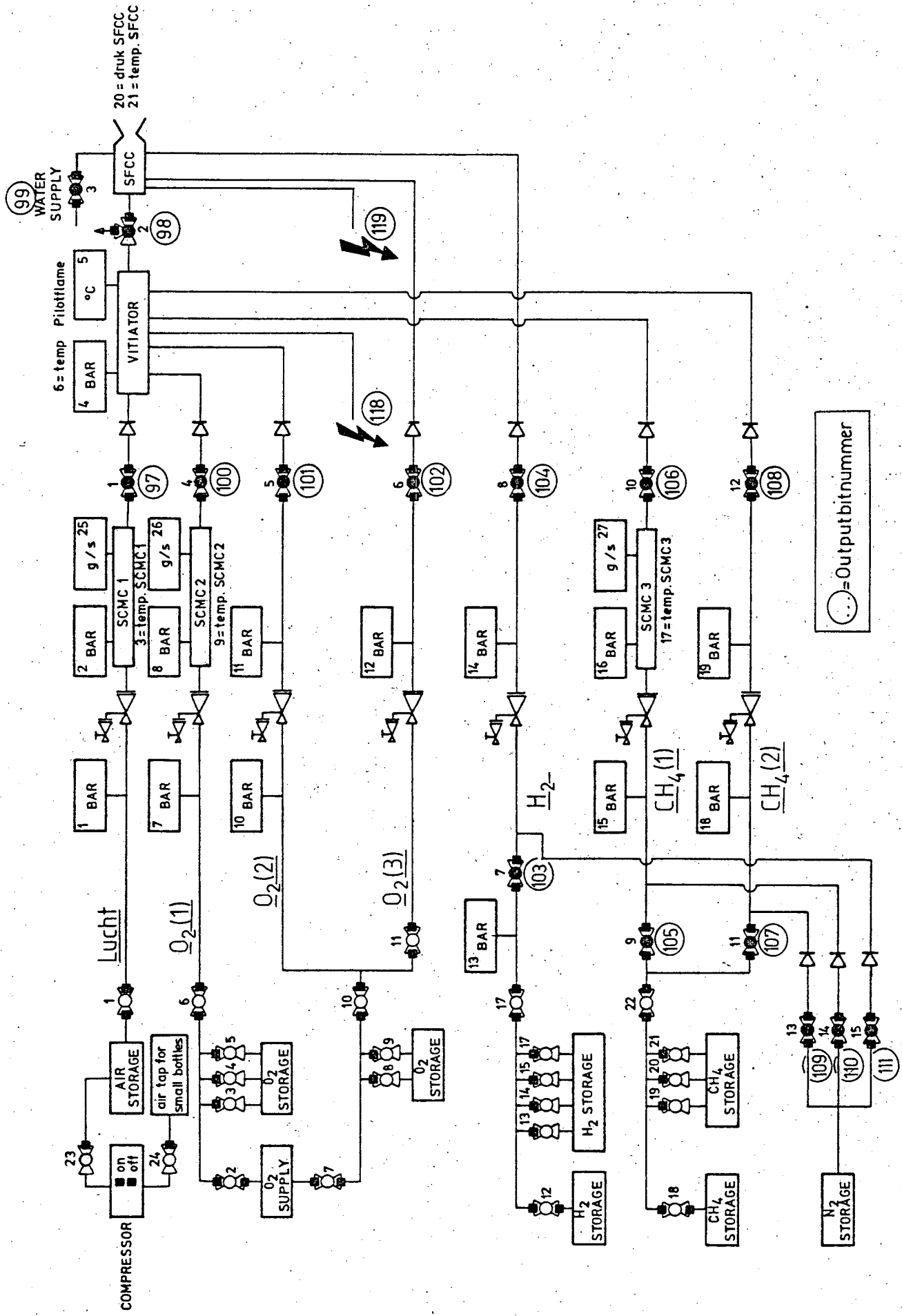
Tabel 1.4. Analoge uitgangen van de microprocessor.

kanaal	connectie
1	massastroom lucht-SMRD
2	massastroom O ₂ -SMRD
3	massastroom CH ₄ -SMRD
4	reserve

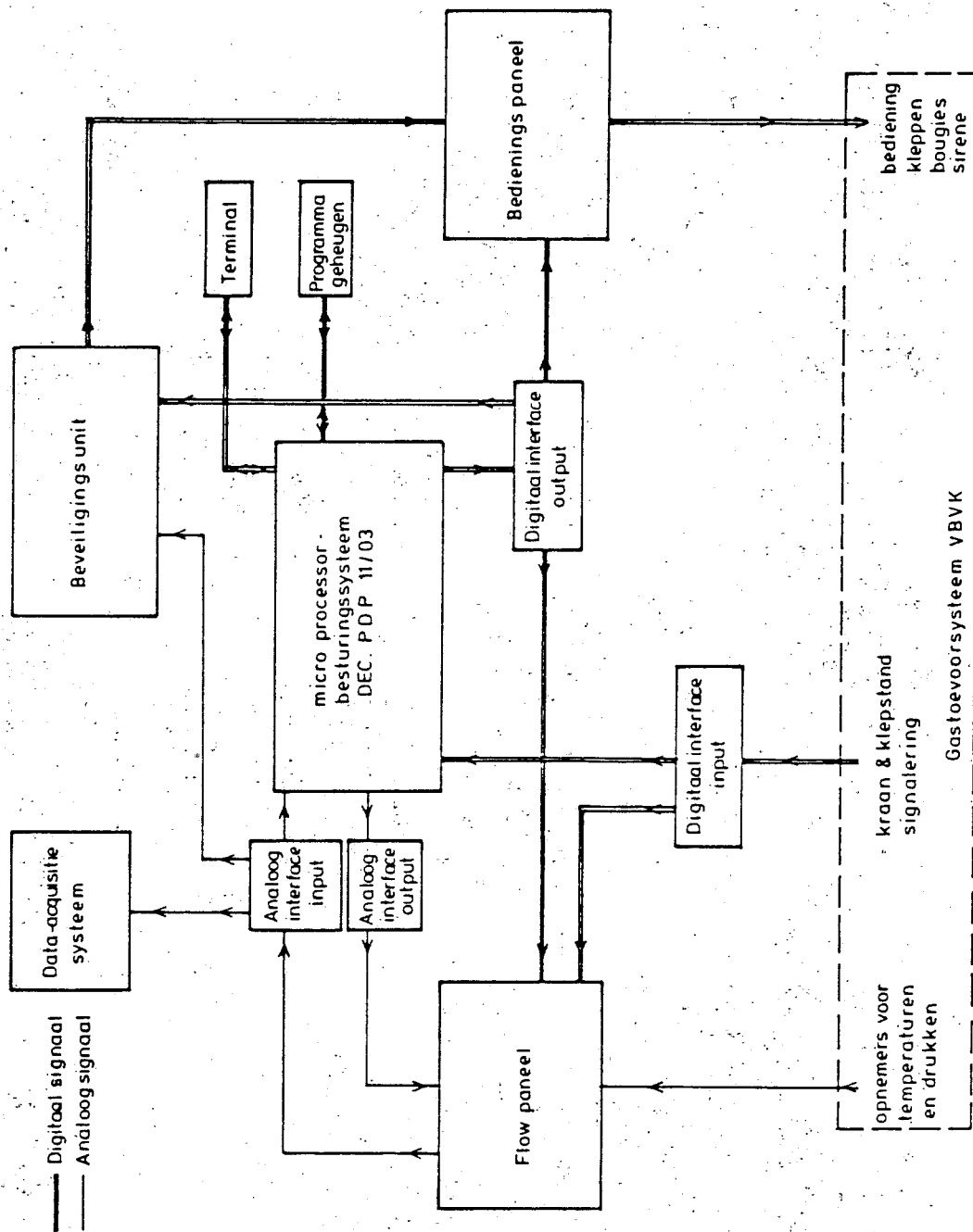
Tabel 1.5. Overzicht van de soort proeven die met het gastoevoersysteem kunnen worden uitgevoerd.

soort proef	code proef
hybride raketmotor	H
VBVK met koude lucht en ontsteking	SCI
VBVK met hete lucht zonder ontsteking	SH
VBVK met hete lucht en ontsteking	SHI
SCI proef met verrijkte lucht	SECI
SH proef met verrijkte lucht	SEH
SHI proef met verrijkte lucht	SEHI
ijking sonische meet- en regeldoorsneden	CAL*)

*) nog niet beschikbaar op het moment van schrijven van dit rapport.



Figuur 1.1. VBVK gasvoersysteem.



Figuur 1.2. Blokschema besturingssysteem gastvoerininstallatie VBVK.

2. DE PROGRAMMATUUR

Het computerprogramma voor de microprocessor van het gastoevoersysteem (SFCC programma) bestaat uit 2 gedeeltes:

1. het eigenlijke SFCC programma;
2. een timing routine (VBVK01).

De timing routine wordt iedere seconde aangeroepen en meteen uitgevoerd. Het 'eigenlijke' programma wordt tijdens het uitvoeren van deze routine onderbroken. Deze routine verzorgt de volgende zaken:

1. afgifte van een 1 Hz puls aan de beveiligingsunit;
2. verrichting van alle i/o, a/d en d/a handelingen, echter niet tijdens de wezenlijke uitvoering van een proef;
3. controle van de bitsetting en eventuele andere belangrijke grootheden;
4. berekening van alle massastromen zodra alle gegevens hiervoor aanwezig zijn.

Het SFCC-programma is opgebouwd uit een hoofdprogramma (VBVK) en een aantal sub-routines (VBVKXX, waarbij een X een cijfer voorstelt). In het programma zijn 3 stadia te onderscheiden:

1. inlezen van diverse grootheden (type proef, e.d.);
2. controle en aanwijzingen voor verbetering van de drukken, kraan/klep instellingen etc., teneinde een proef goed te laten verlopen;
3. uitvoering van de proef.

De volgende subroutines zijn hierbij van belang:

- ad 1: VBVK1X met X = 0, 1
- ad 2: VBVK1X met X = 2, 3, 4, 5, 6, 7, 9
- ad 3: VBVK2X met X = 0

De overige, nog niet genoemde routines zijn:

- VBVK00 - initialisatie
- VBVK02 - interrupt routine bit 0 (startknop); niet in gebruik
- VBVK03 - interrupt routine bit 1 (stopknop); niet in gebruik
- VBVK30 - i/o, a/d en d/a routines; omzetting naar SI eenheden voor de a/d kanalen en omzetting naar Volts voor de d/a kanalen
- VBVK40 - verzorgt (eventueel) aan het einde van een proef een print-out van de opgeslagen drukken, temperaturen en massastromen tijdens een proef
- VBVK99 - error routine
- BLOCK DATA
 - datablok met alle vaststaande gegevens (ijkingen, e.d.).

Een programma-listing is opgenomen als appendix A.

2.1. Flowschema van het SFCC-programma

De aanroep van de diverse routines tijdens de loop door het programma VBVK is gegeven in figuur 2.1. In deze figuur staat ook aangegeven wat op een bepaald moment aan de orde is. In tabel 2.1 staat kort vermeld wat alle subroutines uitvoeren.

2.2. Common blocks in het SFCC-programma

Communicatie tussen de diverse programma-onderdelen geschiedt voornamelijk met behulp van zgn. common blocks. Het is hierdoor mogelijk alle grootheden die de

timing routine binnenhaalt door te geven aan het VBVK-programma. Ook eventuele fouten die de timing routine constateert worden zo doorgegeven aan het hoofd-programma. Belangrijke commonblocks met betrekking tot de timing routine zijn:

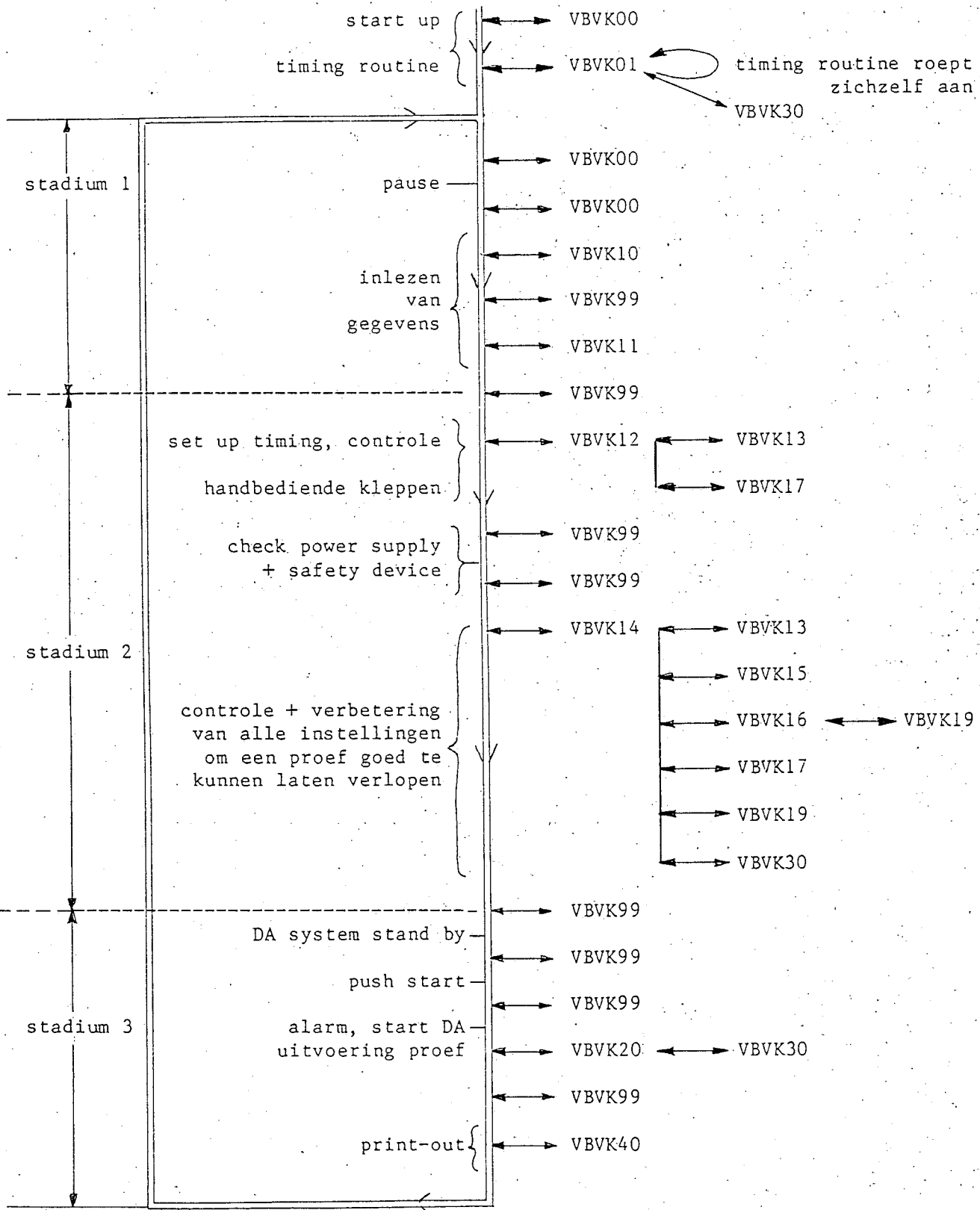
1. het blank common block
2. het common block/gen/
3. het common block/intrp/
- 4 het common block/ddl/

De inhoud van deze common blocks is gegeven in Tabel 2.2. De overige common blocks bevatten de in het BLOCK DATA opgeslagen informatie en gegevens die ingevoerd worden.

Het programma VBLOK geeft een print-out van alle common blocks die in de BLOCK DATA routine worden gebruikt, alsmede een korte toelichting van de betekenis van alle variabelen.

De overige common blocks bbl t/m bb4 zijn gegeven in tabel 2.3.

Figuur 2.1. Overzicht programmaverloop.



⇒ = VBVK-programma

Tabel 2.1. Functie van de gebruikte subroutines.

subroutine	functie
VBVK00	initialisatie
VBVK01	timing routine
VBVK02	interrupt routine start bit
VBVK03	interrupt routine stop bit
VBVK10	inlezen invoergegevens, deel 1
VBVK11	inlezen invoergegevens, deel 2
VBVK12	opzetten van de kleptiming; berekening data-storage tijd; controle handbediende kleppen
VBVK13	controle en/of correctie van de handbediende kleppen al naar gelang de waarde van nfl1
VBVK14	controle en verbetering van drukken teneinde de goede massastromen te verkrijgen
VBVK15	berekening massastromen
VBVK16	berekening van het mogelijke secundaire drukgebied bij een bepaalde massastroom
VBVK17	controle op de stand van de buffervaten
VBVK19	print-out routine voor de verschillende massastroom bepalende apparaten (SMRD, diafragma, aantal windingen)
VBVK20	uitvoering proef; opslag data om de zoveel tijd
VBVK30	i/o routine; omrekening naar SI grootheden
VBVK40	print-out routine van de opgeslagen data tijdens VBVK20
VBVK99	error routine
BLOCK DATA	vaststaande gegevens (zit in de file VBVK99.FOR)

Tabel 2.2. De common blocks met betrekking tot de timing routine.

common	variabele(n)	type	betekenis
/blank/	area (4)	integer *2	wordt gebruikt voor het aanroepen van de itimer routine; deze zorgt voor het aanroepen van de timing routine om de 1 sec, zie Ref. 3.
	iy, im, id	integer	bevat jaar, maand en dag na aanroep van de routine VBVK00.
	it(8)	logical *1	bevat tijd na aanroep van de routine VBVK00.
/gen/	nin(64)	integer	geeft de bitsetting van de 64 input bits nadat VBVK30 is aangeroepen. De waarde van nin(i) zal of 0 of 1 zijn.
	nval(128)	integer	de 1e 64 elementen zijn bedoeld als controle van de inputbits. Deze worden in het programma van zodanige waarden voorzien dat ze gelijk moeten zijn aan nin. Een check hierop gebeurt in de timing routine. De 2e 64 elementen zijn de output bits. De timing routine zorgt ervoor (echter niet tijdens een proef) dat VBVK30 wordt aangeroepen om de outputbits door te geven.
	di(24)	real	bevat informatie van de a/d kanalen. Deze info is reeds geconverteerd (door VBVK30) naar K en Pa.
	do(4)	real	bevat informatie voor de d/a kanalen. Deze is gegeven in kg/s en wordt door VBVK30 geconverteerd naar Volts.
	acti	real	geeft bij uitvoering van een proef aan op welk tijdstip er weer actie (kleppen etc.) ondernomen moet worden.
	ierror	integer	errorflag; deze wordt op zijn waarde getest door VBVK99: 0 = geen fout; negatief = fouten geconstateerd; reageer hierop zonder gebruik te maken van de beveiligingsunit; positief = fout, maak ook gebruik van de beveiligingsunit.
	iflag1	integer	= 0: geen test op correctheid nin met behulp van nval, tevens geen berekening massastromen. ≠ 0: test op correctheid; berekent massastromen. Iflag1 geeft nu het aantal keren aan dat een fout is geconstateerd in nin. Na meer dan 10 keer een fout geconstateerd te hebben wordt ierror gezet.
	iflag2	integer	geeft het aantal keren dat informatie weggeschreven is. Deze informatie kan aan het eind van de proef uitgeprint worden.

Tabel 2.2. Vervolg.

common	variabele(n)	type	betekenis
	iflag3	integer	= 0: geen controle op aan-positie voeding en beveiligingsunit. ≠ 0: wel controle.
	iflag4	integer	= 0: VBVK30 wordt aangeroepen door de timing routine. = 1: VBVK30 wordt aangeroepen door VBVK20 (gedurende de wezenlijke uitvoer van een proef).
/intrp/	iint1	integer	= 0: geen interrupt bit 0. = 1: interrupt bit 0 (start).
	iint2	integer	= 0: geen interrupt bit 1. = 1: interrupt bit 1 (stop).
/ddl/	elap	real	om de elap seconde wordt info weggeschreven naar store (zie beneden); iflag2 houdt het aantal keren bij (iflag2 < 10).
	dmas(7)	real	bevat de berekende massastromen van de gebruikte lijnen.
	store(290)	real	zie elap; wordt in VBVK20 volgeschreven.

Tabel 2.3. De common blocks bbl t/m bb4.

common	variabele(n)	type	betekenis
/bbl/	ntexp	integer	type proef: 1 = H, 2 = SCl, etc.
	num	integer	proefnummer.
	amox	real	gewenste massastroom oxidator (kg/s)
	am(7)	real	gewenste/berekende massastromen voor de 7 lijnen.
	tburn	real	brandtijd (s).
	aprel	real	verwachte druk in VBVK (Pa).
	ncool	integer	0/1 voor geen/wel waterkoeling.
/bb2/	tign	real	ontsteektijd (s).
	tpre	real	aanloofase (s).
	rtemo	real	temperatuur (K) van het gas uit de vitiator.
	enrp	real	verrijgingspercentage (%).
	ndary	integer	0/1 voor geen/wel gebruik data-acquisitiesysteem.
/bb3/	ntim (50)	integer	bevat 'timing points' voor deze proef (zie common bd5).
	ndio(50)	integer	bevat 'valve bit numbers' voor deze proef (zie common bd5).
	ti(10)	real	bevat de echte tijden behorende bij de 'timing points'.
	nsine	integer	totaal uit te voeren operaties.
	nfl1	integer	0/1 voor enkel kijken/kijken en actie door VBVK13 en VBVK17 bij foutstaande handbediende kleppen.
/bb4/	ln(7)	integer	0/1 voor lijn in orde/niet in orde na aanroep VBVK13.
	lm(7)	integer	0/1 voor lijn niet gebruikt, te gebruiken tijdens proef.
	ntest(24)	integer	scratch array met info die ook in nval komt te staan; enkel voor handbediende kleppen.

2.3. Invoer van gegevens

De routines VBVK10 en VBVK11 verzorgen de invoer. Bij de invoer zijn de common blocks bbl en bb2 van belang, daar hier de ingevoerde informatie in komt te staan.

De volgende gegevens worden door VBVK10 gevraagd:

1. type experiment: H, SCI, SH, SH1, SECI, SEH, SEH1 of CAL, de bijbehorende variabele ntext krijgt de waarde 1 voor de H test, 2 voor de SCI test, etc.;
2. testnummer, de bijbehorende variabele is num;
3. benodigde massastroom (kg/s). De bijbehorende variabele is amox;
4. brandtijd (s). De bijbehorende variabele is tburn;
5. de verwachte druk in de VBVK (bar). De bijbehorende variabele is aprel en wordt geconverteerd naar Pascal;
6. geen/wel gebruik van de waterkoeling. De bijbehorende variabele ncool krijgt resp. de waarde 0 of 1.

De overige invoergegevens worden door VBVK11 opgevraagd:

7. geen/wel gebruik van het data-acquisitiesysteem. De bijbehorende variabele ndasy is resp. 0 of 1;
8. ontsteektijd (s) voor resp. de H, SCI, SH1, SECI, SEH1 en CAL test. De bijbehorende variabele is tign;
9. aanlooffase (s) voor resp. de SCI, SECI en CAL test. De bijbehorende variabele is tpre;
10. temperatuur (K) van het gas uit de vitiator voor de SH, SH1, SEH, SEH1 en CAL test. De bijbehorende variabele is rtemo.
11. Verrijgingspercentage (%) voor resp. de SECI, SEH, SEH1 en CAL test. De bijbehorende variabele is enrp.

De volgende tests worden door VBVK11 uitgevoerd:

1. enrp \geq 0; indien $<$ 0 dan foutmelding;
2. $240 \leq$ retemo \leq 1250; indien niet het geval, dan foutmelding.

2.4. Verwerking van de gegevens

Aan de hand van de invoergegevens wordt de precieze timing voor elke proef bepaald. De ontsteektijd, de aanlooffase en de brandtijd (ook wel operationele fase genoemd) spelen daarbij een belangrijke rol. De ontsteektijd is de tijd dat de bougie in de VBVK geactiveerd wordt, waarbij tevens een mengsel van H₂ en O₂ (of lucht) aanwezig is. De aanlooffase is alleen voor de SCI en SECI proef belangrijk. Voor de SCI proef is de aanlooffase de tijd tussen het begin van de ontsteking en het openen van de luchttoevoer, terwijl bij de SECI proef het de tijd is tussen de opening van de luchttoevoer en het begin van de ontsteking. Dit verschil wordt hier gemaakt doordat op het moment van schrijven van dit rapport de driewegklep (of afblaasklep) nog niet beschikbaar is terwijl SCI proeven (in tegenstelling tot SECI proeven) reeds uitgevoerd moeten kunnen worden. Wanneer de driewegklep geïnstalleerd is zal de aanlooffase bij SCI proeven aangepast moeten worden aan die zoals bij SECI proeven gedefinieerd.

De brandtijd van de H en SCI proef is de tijd vanaf het eind van de ontsteking tot het stoppen van de oxydatortoevoer. Voor de overige proeven is het de tijd dat de afblaasklep geopend is. Wanneer de afblaasklep geïnstalleerd is zal ook voor H en SCI proeven de brandtijd gelijk zijn aan de tijd dat de afblaasklep geopend is.

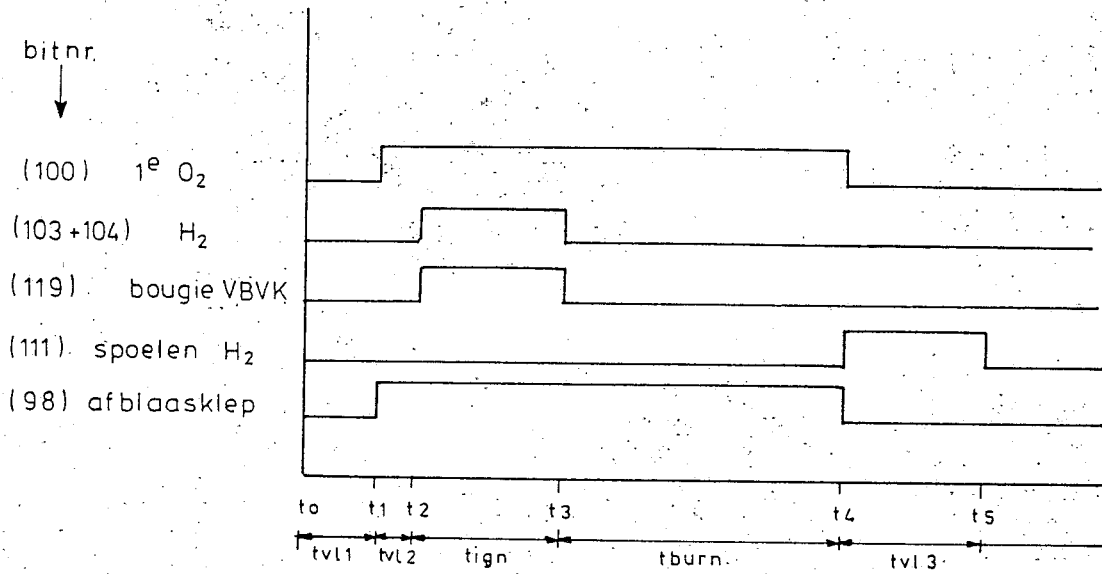
Na de operationele fase worden de leidingen schoongebazen met stikstof (spoil-fase).

De procedures voor de diverse proeven staan vermeld in de Figuren 2.2 t/m 2.6. Voor SEH proeven, zie Figuur 2.2; voor SEHI proeven, zie Figuur 2.5.. Hierbij staat uitgezet op welk tijdstip (t_0 , t_1 , etc.) welke klep of bougie bediend moet worden. Tussen haakjes staat aangegeven welk(e) bitnummer(s) hierbij van belang zijn. Deze vaststaande gegevens zijn opgeslagen in common block bd5 (zie uitdraai van het programma VBLOK).

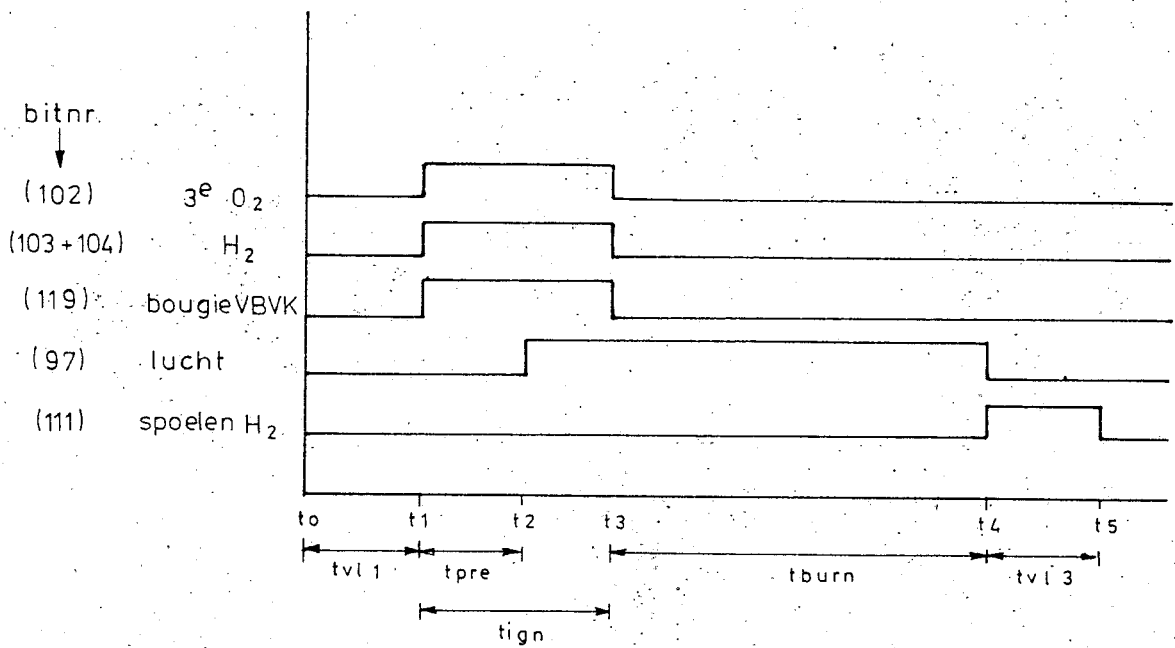
In de routine VBVK12 worden tijden toegekend aan t_0 , t_1 , ... volgens het schema zoals gegeven in Tabel 2.4.

De informatie over de timing wordt opgeslagen in common block bb3. Deze bevat voor een deel dezelfde informatie als common block bd5. Gekeken wordt welke lijnen gebruikt gaan worden (opslag in block bb4) en hoe lang een proef duurt. Hiermee wordt uitgerekend om de hoeveel seconde data-opslag kan plaatsvinden (elap) om de store array compleet te vullen tijdens de proef.

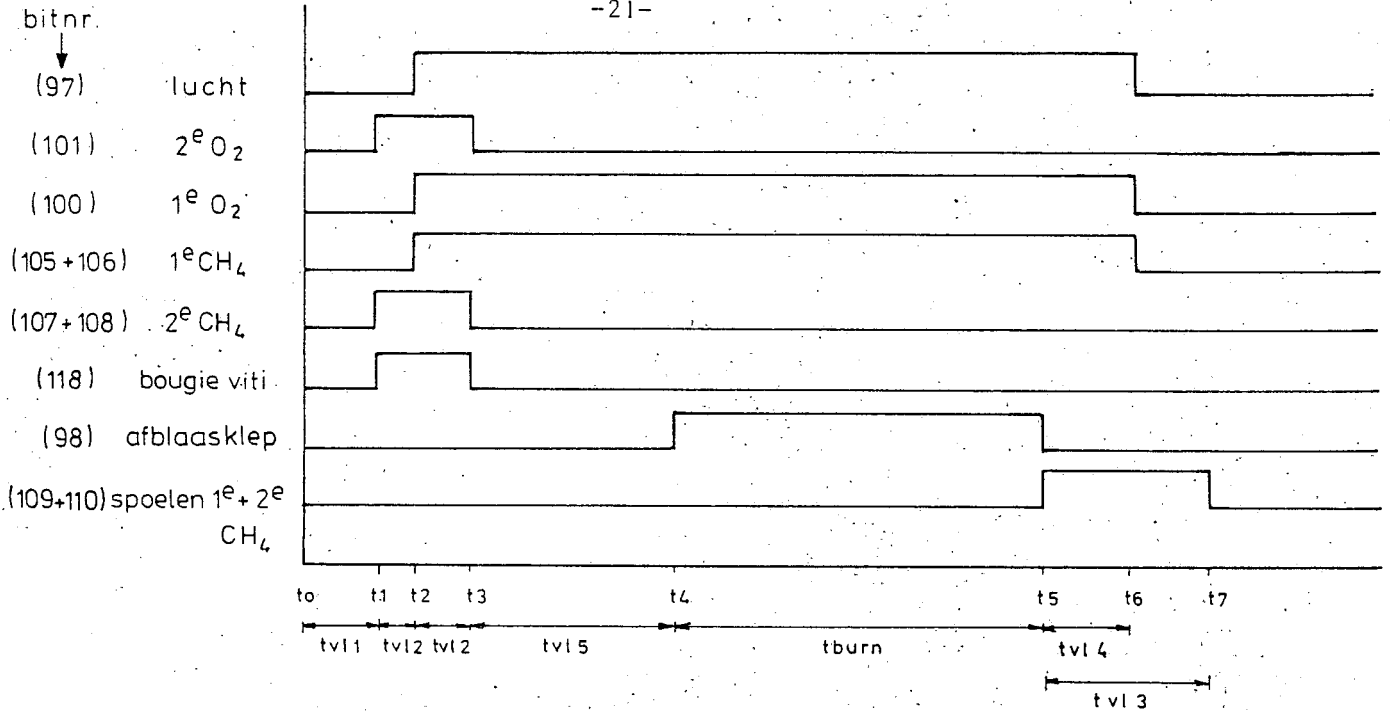
Hierna worden de routines VBVK13 en VBVK17 aangeroepen om te kijken of alle handbediende kleppen reeds goed staan. Voor de lijnen die niet gebruikt worden en waarvan kleppen openstaan vraagt het programma deze te sluiten.



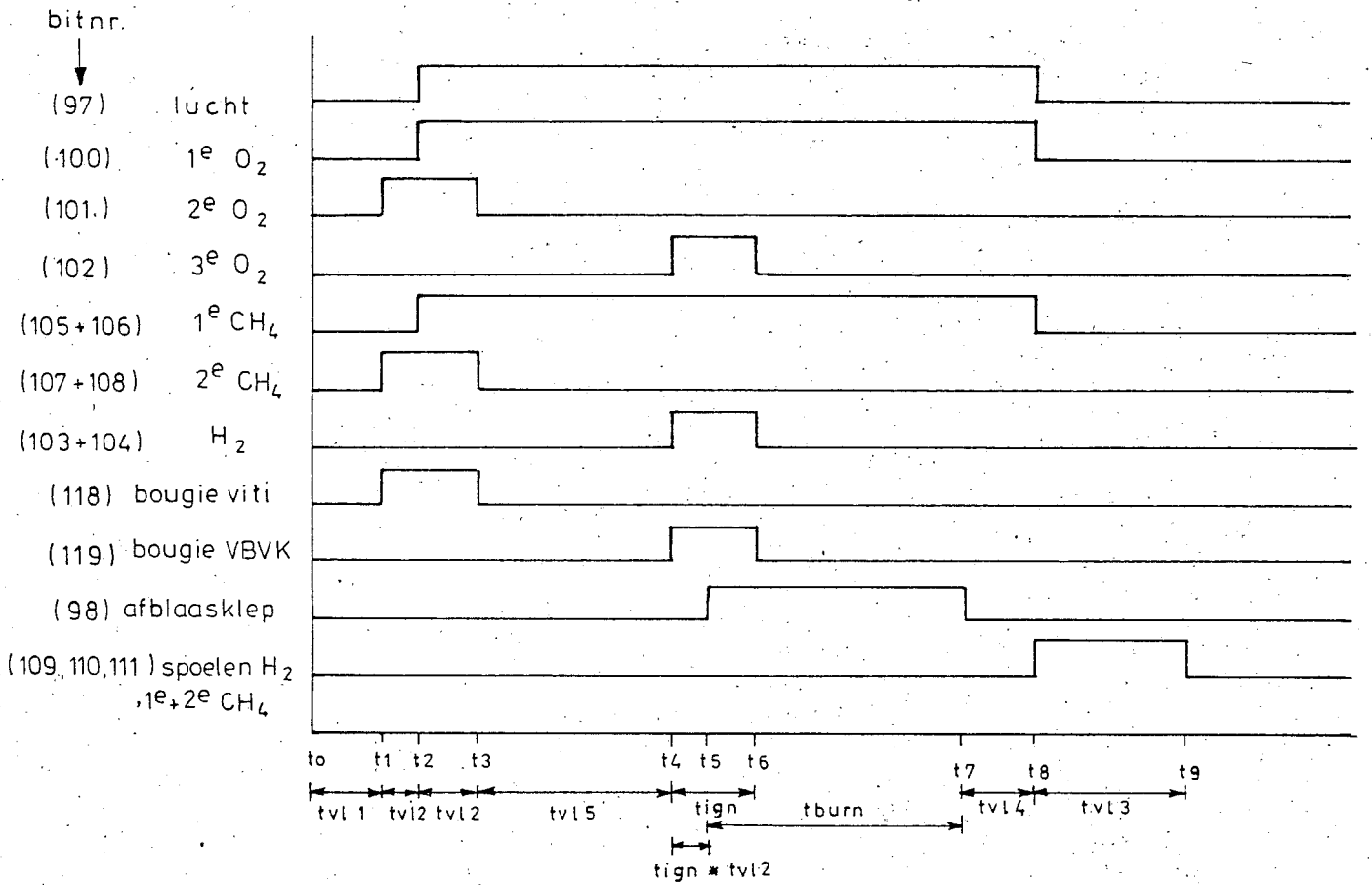
Figuur 2.2. Overzicht van de opeenvolging van gebeurtenissen bij een H proef.



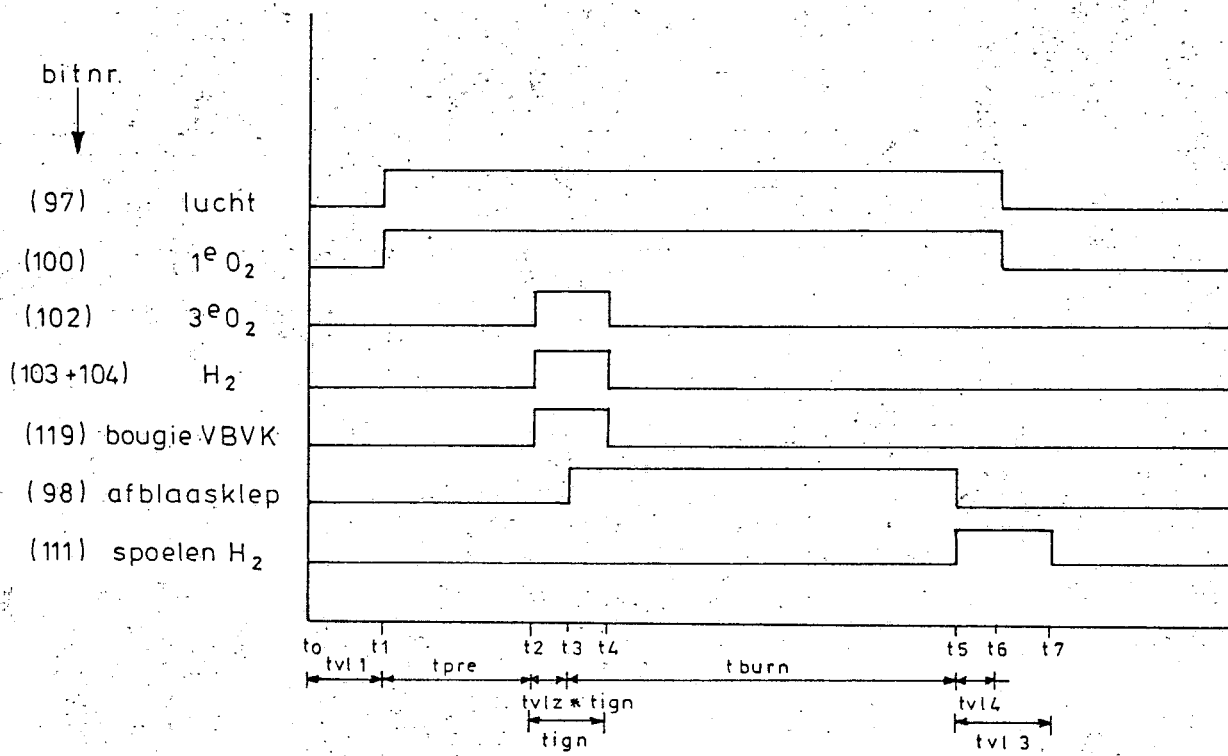
Figuur 2.3. Overzicht van de opeenvolging van gebeurtenissen bij een SCI proef.



Figuur 2.4. Overzicht van de opeenvolging van gebeurtenissen bij een SH proef.



Figuur 2.5. Overzicht van de opeenvolging van gebeurtenissen bij een SHI proef.



Figuur 2.6. Overzicht van de opeenvolging van gebeurtenissen bij een SECI proef.

Tabel 2.4. Toekenning van tijden aan t0, t1 etc. voor de verschillende proeven.

type proef	t0 (s)	t1 (s)	t2 (s)	t3 (s)	t4 (s)	t5 (s)	t6 (s)	t7 (s)	t8 (s)	t9 (s)
H	0	t0+tvλ1	t1+tvλ2	t2+tign	t3+tburn	t4+tvλ3				
SCI	0	t0+tvλ1	t1+tpre	t1+tign	t3+tburn	t4+tvλ3				
SH	0	t0+tvλ1	t1+tvλ2	t2+tvλ2	t3+tvλ5	t4+tburn	t5+tvλ4	t5+tvλ3		
SHI	0	t0+tvλ1	t1+tvλ2	t2+tvλ2	t3+tvλ5	t4+tvλ2*tign	t4+tign	t5+tburn	t7+tvλ4	t8+tvλ3
SECI	0	t0+tvλ1	t1+tpre	t2+tvλ2*tign	t2+tign	t3+tburn	t5+tvλ4	t5+tvλ3		
SEH = SH										
SEH = SHI										

tign = ontsteektijd
 tpre = aanloofase
 tburn = brandtijd

tvλ1 = 1,0 s
 tvλ2 = 0,5 s
 tvλ3 = 15,0 s
 tvλ4 = 2,5 s
 tvλ5 = 30,0 s

2.5. Berekening van de massastromen

In de routine VBVK14 wordt eerst VBVK15 aangeroepen die de massastromen voor alle lijnen uitrekent.

Voor de H test wordt de massastroom van de 1e O₂ lijn gelijk gemaakt aan de ingevoerde waarde amox. Voor de overige proeven² waarbij de vitiator wordt gebruikt, verzorgt de volgende procedure de gewenste massastromen door de SMRD's:

1. bereken de temperatuur van de instromende gassen door het gemiddelde te nemen van de temperaturen in de 3 SMRD's;
2. kijk of er voor deze temperatuur een tabel met polynoomcoëfficiënten aanwezig is in common block bdl, eventueel door lineaire interpolatie tussen 2 tabellen;
3. bereken de relatieve massastromen ϕ volgens de methode van R.G. van Bruggen (SFCC publication 1) (Ref. 4).

$$\phi_{\text{CH}_4} = c_0 + c_1 * r_{\text{temo}} + c_2 * r_{\text{temo}}^2 + c_3 * r_{\text{temo}}^3 + c_4 * r_{\text{temo}}^4$$

ϕ_{CH_4} = relatieve methaan massastroom :

c0 t/m c4: polynoom constanten (zie common bdl)

rtemo = gewenste gastemperatuur (K).

De overige relatieve massastromen volgen hieruit:

$$\phi_{\text{O}_2} = 5.575 * \phi_{\text{CH}_4}$$

$$\phi_{\text{lucht}} = 1 - 6.575 * \phi_{\text{CH}_4}$$

De grootste relatieve fout die gemaakt wordt door gebruik te maken van de polynoombenadering is ca 0,5%.

De overige massastromen worden berekend volgens:

1. H test:

$$m_{\text{H}_2} = \text{flm1} * m_{\text{O}_2} \text{ (1e lijn); met een maximum gelijk aan flm2 kg/s}$$

2. SCI en SECI test:

$$m_{\text{O}_2} \text{ (3e lijn)} = \text{flm3} * m_{\text{lucht}}$$

$$m_{\text{H}_2} = \text{flm1} * m_{\text{O}_2} \text{ (3e lijn); met een maximum gelijk aan flm2 kgs}$$

3. SH en SEH test:

$$m_{\text{O}_2} \text{ (2e lijn)} = \text{flm6}$$

$$m_{\text{CH}_4} \text{ (2e lijn)} = \text{flm4} * m_{\text{O}_2} \text{ (2e lijn)}$$

4. SHI en SEHI test:

$$m_{\text{O}_2} \text{ (2e lijn)} = \text{flm6}$$

$$m_{\text{O}_2} \text{ (3e lijn)} = \text{flm3} * m_{\text{lucht}}$$

$$m_{H_2} = flm1 * m_{O_2} \text{ (3e lijn); met een maximum gelijk aan flm2 kg/s}$$

$$m_{CH_4} \text{ (2e lijn)} = flm4 * m_{O_2} \text{ (2e lijn)}$$

flm1 t/m flm4 en flm6 zijn constanten en worden gegeven in common block bd7.

2.6. Verrijgingspercentage

De massastroom van O_2 door de SMRD wordt vermeerderd met het bedrag $(enrp/100) \times amox$, waarbij $enrp$ het verrijgingspercentage is en $amox$ de gewenste oxidator-massastroom. De berekening van de massastromen van CH_4 , O_2 en lucht die nodig zijn om de gewenste luchttemperatuur te krijgen geschiedt door de relatieve massastromen te vermenigvuldigen met $(1-enrp/100) \times amox$ in plaats van met $amox$. In de praktijk betekent dit dat bij een hoger verrijgingspercentage de gewenste gastemperatuur niet gehaald wordt door afkoeling als gevolg van bijmenging met koude zuurstof.

2.7. Instelling van de drukken en massastromen

Na de berekening van de massastromen worden de elektrische kleppen geopend die het mogelijk maken de secundaire druk in de SMRD's te meten. Om voor elke lijn j de juiste massastroom te krijgen, wordt de volgende procedure uitgevoerd.

1. met de errorflag ne voor deze lijn op nul;
2. invoer van het gebruikte diafragma of penetratielengte of aantal windingen;
3. berekening van de operatietijd voor deze lijn;
4. berekening van het mogelijke secundaire druk gebied door de routine VBVK16. Dit drukgebied wordt bepaald door de gevraagde massastroom. Tevens geeft de routine een print-out over de mogelijke instellingen van de secundaire druk en massastroom bepalende apparaten. De maximale secundaire druk mag flm5 Pa bedragen;
5. controle van de errorflag; indien deze groter is dan 5 wordt gevraagd of de massastroom gewijzigd dient te worden. Na wijziging volgt weer een aanroep naar VBVK16;
6. berekening van het buffervolume;
7. berekening van het drukverloop voor deze lijn tijdens de proef volgens:

$$del(p) = m * t * 8314.34 * T / (M * V)$$

met:

del(p) = drukverloop
m = massastroom
t = operatietijd
T = temperatuur gas
M = molecuulgewicht van het gas
V = buffervolume

8. uitvoering van de volgende tests:
 - a. $del(p) + sec. druk > 0.95 * prim. druk$? Indien ja, dan wordt deze situatie aangeduid met een P;
 - b. sec. druk ligt buiten het secundaire drukgebied, zoals gevonden bij punt 4; deze situatie wordt aangeduid met een S;
 - c. $flm7 * aprel > secundaire druk$ levert eveneens de situatie S op;

- d. het berekende secundaire drukgebied is niet reëel, dat wil zeggen de ondergrens geeft een hogere druk dan de bovengrens; dit levert de situatie '*' op;
9. print-out van diverse gegevens;
10. actie voor de situaties P, S en *:
- a. situatie P: verhoog errorflag met 1.
Wanneer nog niet alle buffervaten van deze lijn geopend zijn wordt gevraagd deze alsnog open te zetten, teneinde de primaire druk te verhogen. Indien alle vaten open staan zal een complete 'start up' procedure voor deze lijn volgen. Na correctie terug naar pt. 5,
 - b. situatie S: verhoog errorflag met 1.
Bij te lage secundaire druk wordt gevraagd deze te verhogen. Bij te hoge secundaire druk dient het reduceerventiel gesloten te worden. Het programma opent dan de bijbehorende elektrische klep om de secundaire druk lager te krijgen dan 1 bar. Na correctie terug naar pt. 5;
 - c. situatie *: verhoog errorflag met 10.
Hierna terug naar pt. 5.

flm5 en flm7 zijn constanten (common bd 7).

Voordat deze procedure uitgevoerd wordt, krijgt de gebruiker nadere informatie door een verkorte procedure uit te voeren. Deze verkorte procedure bevat niet de punten 5 en 10 en test in punt 8 volgens een iets afwijkend schema:

- a) $\text{del}(P) + \text{pl} > 0.95 * \text{prim. druk}$ levert de situatie P. pl is de ondergrens van het secundaire drukgebied zoals gevonden door VBVK16;
- b) secundaire druk buiten het berekende gebied levert de situatie S op;
- c) $2 * \text{aprel} > \text{pl}$ levert de situatie '*' op;
- d) ondergrens hoger dan bovengrens van het berekende secundaire drukgebied geeft de situatie '*'.

Na deze verkorte procedure afgewerkt te hebben, gaat het programma kijken of alle handbediende kleppen goed staan met zonedig correctie (routine VBVK13 en VBVK17). Nadat de volledige procedure goed doorlopen is gaat het 'ready for ... supply' lampje voor de desbetreffende lijn branden. De informatie over alle instellingen die dan beschikbaar is, is voldoende om de juiste massastroom te verkrijgen. Een test op de grootte van de massastroom is echter niet ingebouwd.

Indien de afstelling geregeld is, geeft het programma een overzicht van de relaties die er bestaan tussen de massastroom en de druk en temperatuur van de diverse lijnen. Het is nu mogelijk de proef uit te voeren door op de startknop te drukken. Er wordt eventueel gewacht op het data-acquisitiesysteem. Na het drukken op de startknop wordt alarm gegeven en het data-acquisitiesysteem (indien nodig) gestart.

2.8. Uitvoering van de proef

De uitvoering van de proef wordt verzorgd door de routine VBVK20, die na het drukken op de startknop aangeroepen wordt. Deze routine zet iflag4 (zie tabel 2.2) op 1, zodat de i/o, a/d en d/a kanalen gelezen/geschreven worden in VBVK20 zelf. Er wordt dan niet meer ± 1 seconde gewacht voordat een opdracht doorgegeven wordt door middel van de timing routine.

Om de elap seconde wordt informatie over drukken, temperaturen en massastromen bewaard in de array store. Voortdurend wordt gekeken of het tijdstip al bereikt is om actie te ondernemen. Nadat dit tijdstip bereikt is, worden de desbetreffende output-bits gezet en weggeschreven. Tevens wordt gezorgd voor het branden van de juiste controle-lichtjes.

Optredende fouten worden gesignaleerd door naar de waarde van ierror te kijken en zonodig op een van nul afwijkende waarde te reageren. Dan springt nl. het programma terug naar het hoofdprogramma en roept VBVK99 aan.

2.9. Afwikkeling van de proef

Nadat de proef afgelopen is, worden de output bits op nul gezet (alle kleppen dicht) en wordt gevraagd of de bewaarde informatie uitgeprint dient te worden. Hierna kan weer begonnen worden met het invoeren van gegevens voor de volgende proef.

2.10. Het werken met interrupts

Het geven van een interrupt wordt door het programma beantwoord door het direkt uitvoeren van de zgn. completion routine die bij deze interrupt hoort. De routines die hierbij belangrijk zijn, zijn resp. DJENBL en DJDSBL (par. 4.1) en VBVK02 en VBVK03. Deze laatste 2 routines worden als completion routine gebruikt.

Op dit moment is het nog niet mogelijk (door fouten in DJENBL en de afwezigheid van DJDSBL) van bovenstaande procedure gebruik te maken. In het huidige programma wordt dit ondervangen door naar de input bits 57 en 58 (zie ook tabel 1.1) te kijken. Deze zijn nl. normaal hoog, terwijl in het geval van een interrupt deze laag worden. De controle hierop gebeurt in de routine VBVK01.

2.11. Controle terugmelding elektrische kleppen

In VBVK01 wordt gekeken (zie ook iflag1 par. 2.2, tabel 2.2) of de ingelezen input bits correct zijn. Op dit moment worden enkel de eerste 22 bits op juistheid gecontroleerd. Nadat alle microswitches aangebracht en op functioneren getest zijn kan dit aantal uitgebreid worden tot de eerste 39 bits (zie ook tabel 1.1, par. 1).

3. FOUTMELDINGEN

Een foutmelding wordt signaleerd door een van nul afwijkende waarde voor ierror. Ierror is opgebouwd uit 3 cijfers voorafgegaan door 1 tot 2 cijfers die het nummer van de routine aangeven waarin de fout gedetecteerd is. Een overzicht van de foutmeldingen is gegeven in Tabel 3.1.

Tabel 3.1. Overzicht foutmeldingen.

fout	routine	oorzaak
1001	VBVK01	voedingsspanning afwezig
1002	VBVK01	beveiligingsunit uit
1003	VBVK01	stopknop ingedrukt
1004	VBVK01	meer dan 10 keer werden andere dan verwachte inputbits gelezen (zie dimension nval)
1005	VBVK01	druk in VBVK was 5% groter dan de ingevoerde waarde (aprel)
3001	VBVK03	stop bit geactiveerd (niet in werking)
11001	VBVK11	verrijkspercentage < 0
11002	VBVK11	rtemo < 240 K of rtemo > 1250 K
12001	VBVK12	toegewezen tijden aan t0, t1, etc. zijn niet chronologisch
13001	VBVK13	fout in data.array nv (common bd 6)
14001	VBVK14	geforceerde stop door gebruiker
15001	VBVK15	geforceerde stop door gebruiker
15002	VBVK15	geen tabel voor deze gastemperatuur aanwezig
18001	VBVK20	overflow van de store array (dit kan optreden indien de tijdsvertraging door het uitvoeren van de timing routine meer dan elap sec. bedraagt)

4. OVERIGE PROGRAMMATUUR

De overige programmatuur bestaat uit de programma's VTEST, VALVES en VBLOK.

VTEST is bedoeld om de VBVK computer te testen. Met dit programma kan het volgende gedaan worden:

1. DJINT: aanroep van de initialiseringsroutine DJINT
2. DJIN: aanroep van de routine die de inputbits leest
3. DJOUT: aanroep van de routine die de outputbits schrijft
4. IADC: aanroep van de routine die de A/D kanalen leest
5. DAC: aanroep van de routine die de D/A kanalen schrijft
6. DJENBL: aanroep van de routine die een interrupt bit actief maakt (niet werkend)
7. DJDSBL: aanroep van de routine die een interrupt bit inactief maakt (niet werkend)
8. input I/O test: alle 64 bits worden 2 keer achter elkaar gelezen. Nadat de 1e keer een bit gelezen is kan dit bit omgezet worden en als zodanig de 2e keer uitgelezen worden.
9. output I/O test: alle 64 output bits worden achtereenvolgens hoog
10. verbindingen van D/A kanalen met A/D kanalen: met behulp van deze optie kan direct op het flow paneel een A/D kanaal afgelezen worden

N.B.: VTEST dient bij voorkeur gebruikt te worden indien enkel de simulatieunit aangesloten is.

VALVES is een gestripte versie van het SFCC programma en is geschikt om op een eenvoudige wijze kleppen en bougies te bedienen. Er wordt geen controle op de correctheid van drukken en massastromen uitgevoerd. In totaal kan men 250 handelingen (kleppen e.d.) verrichten.

VBLOK is een programma die een print-out geeft van alle gegevens die in de BLOCK DATA routine van het SFCC programma worden gebruikt. Dit programma kan slechts op de PDP11/45 gedraaid worden. Een voorbeeld van zo'n print-out is gegeven in appendix B.

4.1. Routines voor i/o en a/d, d/a handelingen

De volgende .OBJ files op [118.3] zijn hierbij van belang:

1. PMLRTV.OBJ ... systeemroutines
2. DAC.OBJ ... d/a
3. IADC.OBJ ... a/d
4. DRV11J.OBJ ... i/o

Deze files bevatten o.a. de volgende routines:

1) DJINT

- initialisering: call dkint (iport, io)

Er zijn 8 poorten met elk 16 bits; genummerd 0 t/m 3 voor de 1e DRV11J (input) en 4 t/m 7 voor de 2e DRV11J (output).

Voordat veilig gewerkt kan worden met de DRV11J's (bits ongedefinieerd bij het aanzetten) moet DJINT aangeroepen worden.

iport: 0 7

io: =1 voor output; = 0 voor input

2) DJIN

- input i/o: call (ivalue, iport)

ivalue = octale representatie voor iport (0 ... 7).

Het is mogelijk ook de setting van de output bits te lezen.

3) DJOUT

- output i/o: call djout (ivalue, iport)

ivalue is octale representatie voor iport (4 ... 7)

4) IADC

- a/d input: ivalue = IADC (ichan)

ichan = kanaalnummer 0 ... 63

ivalue = 0 ... 7777 voor -5 V tot +5 V

0 V → "4000"

5) DAC

- d/a output: call dac (ichan, ivalue)

ichan = kanaalnummer (0 ... 3)

ivalue = 0 ... 4095 voor +5 V tot -5 V

0 V = 2048

6) DJENBL en DJDSBL

- interrupt aanzetten: call DJENBL (ibitnr, naam complete external routine).

ibitnr = 0 of 1

- interrupt inactief maken: call DJDSBL (ibitnr)

Vervangende testroutines zijn te vinden in IODUM.FOR op [118,3].

5. BESCHIKBAARHEID VAN DE PROGRAMMA'S

De programma's staan in principe op het netwerk van het PML. Sav versies kunnen aangemaakt worden door onder [118,3] de volgende procedures uit te voeren:

1. voor VTEST : VTEST.CMD
2. voor VALVES: VALVES.CMD
3. voor SFCC : SFCC1.CMD en SFCC2.CMD
4. voor VBLOK : VBLOK.CMD

De versie van SFCC die slechts op de PDP11/45 kan draaien ter controle heet VBVK en kan aangemaakt worden door in plaats van SFCC2.CMD, SFCC3.CMD uit te voeren.

Het verdient aanbeveling na wijziging van een programma ook de geprinte update datum te veranderen welke in het hoofdprogramma is te vinden.

N.B. De block data zijn opgenomen in de file VBVK99.FOR.

6. REFERENTIES

1. P. Korting, A. van Lingen en J. Versmissen.
'Een regel/bedieningsysteem voor de gastoevoerinstallatie van de Vaste Brandstof Verbrandingskamer', SFCC publikatie no. 3, Rapport LR-366/PML 1983-101, Afdeling der Luchtvaart- en Ruimtevaarttechniek, Technische Hogeschool Delft/Prins Maurits Laboratorium TNO, Delft/Rijswijk, januari 1983.
2. H. Albers, P. Korting en J. Versmissen.
'Technische gebruiksaanwijzing voor het regel/bedieningssysteem van de gastoevoerinstallatie voor de Vaste Brandstof Verbrandingskamer', SFCC publikatie no. 8, Rapport LR-410/PML 1983-169, Afdeling der Luchtvaart- en Ruimtevaarttechniek, Technische Hogeschool Delft/Prins Maurits Laboratorium TNO, Delft/Rijswijk, januari 1984.
3. An, 'RT-11 Programmer's Reference Manual', AA-H378B-TC, Digital Equipment Corporation, Maynard, Massachusetts, March 1983.
4. R.G. van Bruggen.
'Luchtverhitting door middel van een vitiator', SFCC publikatie no. 1. Memorandum M-435/Rapport PML 1982-147, Afdeling der Luchtvaart- en Ruimtevaarttechniek, Technische Hogeschool Delft/Prins Maurits Laboratorium TNO, Delft/Rijswijk, oktober 1982.

APPENDIX A

PROGRAMMA LISTINGS

PROGRAM VBVK

C VBVK IS WRITTEN BY :

C*****

C* R.J. VAN DER WAL *

C* TNO-PML *

C* LANGE KLEIWEG 137-139 *

C* RIJSWIJK *

C* THE NETHERLANDS *

C*****

COMMON AREA, IY, IM, ID, IT

COMMON/BB2/TIGN, TPRE, RTEMO, ENRP, NDASY

COMMON/DD1/ ELAP, DMAS(7), STORE(290)

COMMON/GEN/NIN(64), NVAL(128), DI(24), DO(4), ACTI,

\$ IERROR, IFLAG1, IFLAG2, IFLAG3, IFLAG4

COMMON/INTRP/IINT1, IINT2

EXTERNAL VBVK01, VBVK02, VBVK03

INTEGER*2 AREA(4)

LOGICAL*1 IT(8)

C VBVK, A PROGRAM FOR THE GAS-FLOW SYSTEM

C INITIALISE BEFORE USING THE PROGRAM

CALL VBVK00

C ASK FOR A QUEUE ELEMENT

C CALL BACKGROUND ROUTINE, WHICH PERIDICALLY RESCHEDULES ITSELF

IF(IQSET(100).NE.0) STOP 'NO QUEUE ELEMENTS'

CALL ITIMER(0,0,1,0,AREA,1,VBVK01)

1 CONTINUE

C INITIALISE SYSTEM

CALL VBVK00

TYPE 1000,IT, ID, IM, IY

C LEAVE SYSTEM IN PAUSE MODE, BACKGROUND ROUTINE RUNNING

PAUSE

C INITIALISE SYSTEM (CAN BE MUCH LATER THAN FIRST CALL)

CALL VBVK00

TYPE 1000,IT, ID, IM, IY

C TEST..TEST

D CALL VBVK30(1)

D CALL VBVK30(3)

```
C TEST..TEST
C READ INPUT VARIABLES
C SET BIT 69 (TEST FASE)
  NVAL(69) = 1
  CALL VBVK10
C ERROR CONTROL
  CALL VBVK99(I)
  IF(I.GT.0) GOTO 5
C CHECK FOR CONTINUING
  TYPE 1010
  ACCEPT 999,CHAR
  IF(CHAR.NE.4H ) GOTO 5
  CALL VBVK11
C ERROR CONTROL
  CALL VBVK99(I)
  IF(I.GT.0) GOTO 5
C CHECK FOR CONTINUING
  TYPE 1010
  ACCEPT 999,CHAR
  IF(CHAR.NE.4H ) GOTO 5
C SETTINGS FOR VALVES
  CALL VBVK12
C ERROR CONTROL
  CALL VBVK99(I)
  IF(I.GT.0) GOTO 5
C CHECK FOR CONTINUING
  TYPE 1010
  ACCEPT 999,CHAR
  IF(CHAR.NE.4H ) GOTO 5
C SWITCH ON POWER SUPPLY AND SECURITY DEVICE
  IF(NIN(42).EQ.0) TYPE 1004
  IF(NIN(43).EQ.0) TYPE 1006
50 CONTINUE
  CALL VBVK99(I)
  IF(I.GT.0) GOTO 5
  IF(NIN(42).EQ.0) GOTO 50
  IF(NIN(43).EQ.0) GOTO 50
```

```
IFLAG3 = 1
C TEST FOR CORRECTNESS MASS FLOW
  CALL VBVK14
C ERROR CONTROL
  CALL VBVK99(I)
  IF(I.GT.0) GOTO 5
C RESET BIT 69
  NVAL(69) = 0
C CHECK FOR CONTINUING
  TYPE 1010
  ACCEPT 999,CHAR
  IF(CHAR.NE.4H ) GOTO 5
C IGNORE D.A. SYSTEM FOR NDASY = 0
  IF(NDASY.EQ.0) GOTO 20
C WAIT FOR D.A. SYSTEM TO CONTINUE
  IF(NIN(41).EQ.0) TYPE 1008
10 CONTINUE
  CALL VBVK99(I)
  IF(I.GT.0) GOTO 5
  IF(NIN(41).NE.1) GOTO 10
C D.A. SYSTEM STANDBY, SET INDICATOR
  NVAL(122) = 1
  20 CONTINUE
C GASSUPPLY SYSTEM STAND BY, SET INDICATOR
  NVAL(121) = 1
C START VBVK AFTER PUSHING THE START KNOB
C FIRST RESET IINT1
  IINT1 = 0
C TEST..TEST
D   IINT1 = 1
C TEST..TEST
C GIVE A MESSAGE TO PUSH THE START BUTTON
  TYPE 1020
  25 CONTINUE
  CALL VBVK99(I)
  IF(I.GT.0) GOTO 5
  IF(IINT1.EQ.0) GOTO 25
```

```
C SET BIT 70 (PREPARATION FASE)
  NVAL(70) = 1
C GIVE A 5 SECONDS ALARM
  T1 = SECNDS(0.0)
  NVAL(120) = 1
15 CONTINUE
  IF(SECNDS(T1).LT.5.0) GOTO 15
  NVAL(120) = 0
C WAIT UNTIL THE ALARM DIES (APPROX. 30 SEC)
17 CONTINUE
  IF (SECNDS(T1) .LT. 12.0) GOTO 17
C ACTIVATE D.A. SYSTEM, THIS TAKES 20 SECONDS + 1 SEC TIMING ROUTINE
C + 2 SECONDS TO BE SURE THAT THE SIGNAL IS ON MAGTAPE
  NVAL(123) = 1
19 CONTINUE
  IF(SECNDS(T1).LT.35.0) GOTO 19

C DO THE EXPERIMENT
  CALL VBVK20

C RESET IFLAG4 (VBVK30 CALL IN VBVK01)
  IFLAG4 = 0
C ERROR CONTROL
  CALL VBVK99(I)
5 CONTINUE
C ZERO FLAGS AND OUTPUT BITS
  IFLAG1 = 0
  IFLAG3 = 0
  DO 22 J=1,64
    NVAL(J+64) = 0
22 CONTINUE
C GIVE A PRINT-OUT OF THE ERROR CODE
  IF(IERROR.NE.0) TYPE 1030,IABS(IERROR)
C CANCEL ANY BACKGROUND REQUEST (NOT IN USE...)
C   CALL ICMKT(1,AREA)
C CALL FOR VBVK40 IF IFLAG2.NE.0
  IF(IFLAG2.LE.0) GOTO 1
```

```
C ASK IF THERE HAS TO BE OUTPUT
  TYPE 1040
  ACCEPT 999,CHAR
  IF(CHAR.NE.4HY ) GOTO 1
  CALL VBVK40
C NEXT EXPERIMENT
  GOTO 1
  999 FORMAT(A4)
  1000 FORMAT(1H0,'PROGRAM SFCC (UPDATED 8-5-84) AT ',
    $      8A1,' ON ',I2,'-',I2,'-',I2)
  1004 FORMAT(1H0,'SWITCH ON POWER SUPPLY')
  1006 FORMAT(1H0,'SWITCH ON SECURITY DEVICE')
  1008 FORMAT(1H0,'WAITING FOR THE DATA-ACQUISITION SYSTEM',
    $      ' TO CONTINUE')
  1010 FORMAT(/,1H$, 'GIVE STOP/RETURN TO STOP/CONTINUE * ')
  1020 FORMAT(1H0,'PUSH START BUTTON TO CONTINUE',/)
  1030 FORMAT(1H0,'FATAL ERROR ',I6,' DETECTED DURING EXECUTION')
  1040 FORMAT(/,1H$, 'DO YOU WANT OUTPUT OF THE PRESSURES,',
    $      ' ETC. DURING THE TEST (Y OR N) ? * ')
  END
```

SUBROUTINE VBVK00

```
C ROUTINE WHICH INITIALISES SEVERAL VARIABLES
  COMMON AREA,IY,IM,ID,IT
  COMMON/BD4/ TNUL(7),PL(3,7),Z1(7),Z2(7),Z3(7),Y1(7),Y2(7)
  COMMON/DD1/ ELAP,DMAS(7),STORE(290)
  COMMON/GEN/NIN(64),NVAL(128),DI(24),DO(4),ACTI,
  $      IERROR,IFLAG1,IFLAG2,IFLAG3,IFLAG4
  COMMON/INTRP/IINT1,IINT2
  INTEGER*2 AREA(4)
  LOGICAL*1 IT(8)
  DO 5 I=1,4
  DO(I) = 0.0
  5 CONTINUE
  IERROR = 0
```



```
IFLAG1 = 0
IFLAG2 = 0
IFLAG3 = 0
IFLAG4 = 0
IINT1 = 0
IINT2 = 0
DO 10 I=1,64
  NVAL(I) = 0
  NVAL(I+64) = 0
  NIN(I) = 0
10 CONTINUE
  DO 12 I=1,24
    DI(I) = 0.0
  12 CONTINUE
C MAKE PL(1,I) NEGATIVE AND ZERO DMAS
  DO 14 I=1,7
    PL(1,I) = -999.0
    DMAS(I) = 0.0
  14 CONTINUE
C CHECK POWER AND SECURITY UNIT
  NVAL(42) = 1
  NVAL(43) = 1
C INITIALISE DRV11J AND PUT ON INTERRUPT REQUEST
  DO 15 I=1,4
    CALL DJINT(I-1,0)
  15 CONTINUE
    CALL DJINT(4,1)
    CALL DJINT(6,1)
    CALL DJINT(7,1)
C CLEAR PORT B LATER, BECAUSE BIT 89 IS IN IT
  CALL DJINT(5,1)
C ASK DATE AND TIME
  CALL IDATE(IM, ID, IY)
  CALL TIME(IT)
  RETURN
  END
```

```
SUBROUTINE VBVK01(IE)
COMMON AREA, IY, IM, ID, IT
COMMON/GEN/NIN(64), NVIN(64), NVOU(64), DI(24), DO(4), ACTI,
$      IERROR, IFLAG1, IFLAG2, IFLAG3, IFLAG4
COMMON/BB1/NTEXP, NUM, AMOX, AM(7), TBURN, APREL, NCOOL
COMMON/BD3/ AMARK1(7), AMARK2(7), NPP(7), NPS(7), NTT(7)
COMMON/BD4/ TNUL(7), PL(3,7), Z1(7), Z2(7), Z3(7), Y1(7), Y2(7)
COMMON/DD1/ ELAP, DMAS(7), STORE(290)
COMMON/INTRP/IINT1, IINT2
INTEGER*2 AREA(4)
LOGICAL*1 IT(8)
EXTERNAL VBVK01
C READ I/O BITS NIN
      IF(IFLAG4.EQ.0) CALL VBVK30(1)
C ACTIVATE CONTROL SIGNAL FOR SAFE CIRCUIT
C GIVE A SMALL PULSE WITH A FREQUENCY OF 1 HZ
C THIS IS THE ONLY TIME CALLING DJIN AND DJOUT DIRECTLY
C FROM A SUBPROGRAM (EXCEPT OF COURSE VBVK30)
C READ OUTPUT BITS
      CALL DJIN(IVALUE,7)
C COMPLEMENT IVALUE BECAUSE OF THE HARDWARE CONVERSION OF OUTPUT BITS
C SWITCH ON BIT 14
      IVALUE = (.NOT.IVALUE).OR."20000
C WRITE THIS OUTPUT PORT
      CALL DJOUT(IVALUE,7)
C READ A/D INPUT
      IF(IFLAG4.EQ.0) CALL VBVK30(3)
C CHECK POWER SUPPLY AND SECURITY DEVICE
      IF(IFLAG3.EQ.1.AND.NIN(42).EQ.0.AND.IERROR.EQ.0)
$      IERROR = 1001
      IF(IFLAG3.EQ.1.AND.NIN(43).EQ.0.AND.IERROR.EQ.0)
$      IERROR = 1002
C CHECK INTERRUPT FLAGS
C START BIT
      IF(NIN(57).EQ.0) IINT1 = 1
```

```
C STOP BIT
  IF(NIN(58).EQ.0.AND.IERROR.EQ.0) IERROR = 1003
  IF(IFLAG1-1) 10,20,20
  20 CONTINUE
C COMPARE NIN AND NVIN
  DO 100 I=1,22
  IF(NIN(I).NE.NVIN(I)) GOTO 110
100 CONTINUE
  IFLAG1 = 0
110 IFLAG1 = IFLAG1 + 1
  IF(IFLAG1.GT.10.AND.IERROR.EQ.0) IERROR = 1004
C CHECK PRESSURE IN SFCC
  IF(DI(20).GE.1.05*APREL.AND.IERROR.EQ.0) IERROR = 1005
  10 CONTINUE
C CALCULATE MASS FLOWS FOR PANELS
  DO 200 I=1,7
C IGNORE NON USED LINES
  IF(PL(1,I).LT.0.0) GOTO 200
  DMAS(I) = (Z1(I)*PL(1,I)*PL(1,I)+Z2(I)*PL(1,I)+Z3(I)) *
$      (Y1(I)*DI(NPS(I))+Y2(I))
C CORRECT FOR TEMPERATURE IN CASE OF SCMC
  IF(TNUL(I).EQ.0.0) GOTO 200
  DMAS(I) = DMAS(I) * SQRT(TNUL(I)/DI(NTT(I)))
200 CONTINUE
  DO(1) = DMAS(1)
  DO(2) = DMAS(2)
  DO(3) = DMAS(6)
C WRITE D/A OUTPUT
  IF(IFLAG4.EQ.0) CALL VBVK30(4)
C RESET CONTROL SIGNAL FOR SAFE CIRCUIT AND WRITE I/O BITS
C READ OUTPUT BITS
  CALL DJIN(IVALUE,7)
C COMPLEMENT IVALUE BECAUSE OF THE HARDWARE CONVERSION OF OUTPUT BITS
C SWITCH OFF BIT 14
  IVALUE = (.NOT.IVALUE).AND(.NOT:"20000)
C WRITE THIS OUTPUT PORT
  CALL DJOUT(IVALUE,7)
```

```
IF(IFLAG4.EQ.0) CALL VBVK30(2)
CALL ITIMER(0,0,1,0,AREA,1,VBVK01)
RETURN
END
```

```
SUBROUTINE VBVK02
COMMON/INTRP/IINT1,IINT2
C INTERRUPT ROUTINE BIT 0
C START VBVK
C DISABLE INTERRUPT
CALL DJDSBL(0)
IINT1 = 1
RETURN
END
```

```
SUBROUTINE VBVK03
C INTERRUPT ROUTINE BIT 1
C STOP VBVK
COMMON/GEN/NIN(64),NVAL(128),DI(24),DO(4),ACTI,
$ IERROR,IFLAG1,IFLAG2,IFLAG3,IFLAG4
COMMON/INTRP/IINT1,IINT2
C DISABLE INTERRUPT
CALL DJDSBL(1)
C ACTIVATE SECURITY SYSTEM
NVAL(75) = 1
NVAL(125) = 1
IINT2 = 1
IF(IERROR.EQ.0) IERROR = 3001
RETURN
END
```

```

SUBROUTINE VBVK10
C ROUTINE READING INPUT
COMMON/GEN/NIN(64),NVAL(128),DI(24),DO(4),ACTI,
$      IERROR,IFLAG1,IFLAG2,IFLAG3,IFLAG4
COMMON/BB1/NTEXP,NUM,AMOX,AM(7),TBURN,APREL,NCOOL
DIMENSION TYP(8)
NCOOL = 0
DATA TYP/4HH ,4HSCI ,4HSH ,4HSHI ,4HSECI,4HSEH ,
$      4HSEHI,4HCAL /
10 TYPE 1000
1000 FORMAT(/,1H$, 'TYPE EXPERIMENT (H,SCI,SH,SHI,SECI,SEH,SEHI',
$      ' OR CAL) ? * ')
ACCEPT 1001,TEXP
1001 FORMAT(A4)
DO 20 NTEXP=1,8
IF(TEXP.EQ.TYP(NTEXP)) GOTO 30
20 CONTINUE
GOTO 10
30 CONTINUE
TYPE 1010
1010 FORMAT(/,1H$, 'TEST NUMBER ? * ')
ACCEPT *,NUM
TYPE 1020
1020 FORMAT(/,1H$, 'REQUIRED MASS FLOW OXIDIZER (KG/S) ? * ')
ACCEPT *,AMOX
TYPE 1030
1030 FORMAT(/,1H$, 'REQUIRED BURNING TIME FUEL GRAIN (S) ? * ')
ACCEPT *,TBURN
TYPE 1040
1040 FORMAT(/,1H$, 'ESTIMATED PRESSURE LEVEL IN SFCC (BAR) ? * ')
ACCEPT *,APREL
APREL = APREL * 1.0E+05
15 TYPE 1050
1050 FORMAT(/,1H$, 'WATER COOLING (Y OR N) ? * ')
ACCEPT 1051,CHAR
1051 FORMAT(A1)
IF(CHAR.EQ.1HY.OR.CHAR.EQ.1HN) GOTO 25
```

```
GOTO 15
25 CONTINUE
  IF(CHAR.EQ.1HY) NCOOL = 1
  TYPE 2000, TEXP, NUM, AMOX, TBURN, APREL*1.0E-05, CHAR
2000 FORMAT(//,
$      1H0, 'TYPE EXPERIMENT', 25X, A4, /,
$      1H0, 'TEST NUMBER', 29X, I4, /,
$      1H0, 'REQUIRED MASS FLOW OXIDIZER', 13X, F10.5, ' KG/S', /,
$      1H0, 'REQUIRED BURNING TIME FUEL GRAIN', 8X, F10.5, ' S', /,
$      1H0, 'ESTIMATED PRESSURE LEVEL IN SFCC', 8X, F10.5, ' BAR', /,
$      1H0, 'WATER COOLING', 27X, A1, /)
  TYPE 2010
2010 FORMAT(/, 1H$, 'INPUT OK (Y OR N) ? * ')
  ACCEPT 1051, CHAR
  IF(CHAR.EQ.1HN) GOTO 10
  IF(CHAR.NE.1HY) GOTO 10
  RETURN
  END
```

```
  SUBROUTINE VBVK11
C ROUTINE FOR ADDITIONAL INFORMATION
  COMMON/GEN/NIN(64), NVAL(128), DI(24), DO(4), ACTI,
$      IERROR, IFLAG1, IFLAG2, IFLAG3, IFLAG4
  COMMON/BB1/NTEXP, NUM, AMOX, AM(7), TBURN, APREL, NCOOL
  COMMON/BB2/TIGN, TPRE, RTEMO, ENRP, NDASY
C FOR PUR O2 ENRP IS EQUAL TO INFINITIVE, FOR PURE AIR
C ENRP IS EQUAL TO ZERO.
C INITIALIZE
  1 TIGN = 1.0E+30
  TPRE = 1.0E+30
  RTEMO = 1.0E+30
  ENRP = 1.0E+30
C ZERO MASS FLOWS
  DO 3 I=1,7
  AM(I) = 0.0
```

```
3 CONTINUE
  NDASY = 0
C SET NDASY TO 1 IF DATA-ACQUISITION SYSTEM USED
  70 TYPE 1500
1500 FORMAT(
  *      'OUSE OF DATA-ACQUISITION SYSTEM (WHICH CHECKS THE TEST
ROOM)',/
  *      '$(Y OR N) ? * ')
  ACCEPT 1501,CHAR
1501 FORMAT(A1)
  IF(CHAR.EQ.1HY.OR.CHAR.EQ.1HN) GOTO 75
  GOTO 70
  75 CONTINUE
  IF(CHAR.EQ.1HY) NDASY = 1
C SWITCH TO REQUIRED EXPERIMENT
  GOTO (100,101,102,103,104,105,106,107),NTEXP
C H TEST
  100 TYPE 1000
1000 FORMAT(/,1H$, 'IGNITION TIME (S) ? * ')
  ACCEPT *,TIGN
  GOTO 10
C SCI TEST
  101 TYPE 1000
  ACCEPT *,TIGN
  TYPE 1010
1010 FORMAT(/,1H$, 'PREAMBLE FASE (S) ? * ')
  ACCEPT *,TPRE
  GOTO 10
C SH TEST
  102 TYPE 1020
1020 FORMAT(/,1H$, 'REQUIRED TEMPERATURE LEVEL OXIDIZER (K) ? * ')
  ACCEPT *,RTEMO
  GOTO 10
C SHI TEST
  103 TYPE 1000
  ACCEPT *,TIGN
  TYPE 1020
```

ACCEPT *,RTEMO

GOTO 10

C SECI TEST

104 TYPE 1000

ACCEPT *,TIGN

TYPE 1010

ACCEPT *,TPRE

TYPE 1030

1030 FORMAT(/,1H\$, 'ENRICHMENT PERCENTAGE I ? * ')

ACCEPT *,ENRP

GOTO 10

C SEH TEST

105 TYPE 1020

ACCEPT *,RTEMO

TYPE 1030

ACCEPT *,ENRP

GOTO 10

C SEHI TEST

106 TYPE 1000

ACCEPT *,TIGN

TYPE 1020

ACCEPT *,RTEMO

TYPE 1030

ACCEPT *,ENRP

GOTO 10

C CAL TEST

107 TYPE 1000

ACCEPT *,TIGN

TYPE 1010

ACCEPT *,TPRE

TYPE 1020

ACCEPT *,RTEMO

TYPE 1030

ACCEPT *,ENRP

10 CONTINUE

C TEST AND GIVE A PRINT-OUT

TYPE 1999,CHAR


```
      IF(TIGN.EQ.1.0E+30) GOTO 201
      TYPE 2000,TIGN
      GOTO 202
201  TIGN = 0.0
202  IF(TPRE.EQ.1.0E+30) GOTO 203
      TYPE 2001,TPRE
      GOTO 204
203  TPRE = 0.0
204  IF(RTEMO.EQ.1.0E+30) GOTO 205
      TYPE 2002,RTEMO
      GOTO 206
205  RTEMO = -999.0
206  IF(ENRP.GT.1.0E+20) GOTO 207
      TYPE 2003,ENRP
      GOTO 208
207  ENRP = 0.0
208  CONTINUE
1999 FORMAT(1H0,'USE OF DATA-ACQUISITION SYSTEM',19X,A1)
2000 FORMAT(1H0,'IGNITION TIME',27X,F10.5,' S')
2001 FORMAT(1H0,'PREAMBLE TIME',27X,F10.5,' S')
2002 FORMAT(1H0,'REQUIRED TEMPERATURE LEVEL OXIDIZER',5X,F10.5,' K')
2003 FORMAT(1H0,'ENRICHMENT PERCENTAGE',19X,F10.5,'')
      TYPE 2010
2010 FORMAT(/,1H$, 'INPUT OK (Y OR N) ? * ')
      ACCEPT 2011,CHAR
2011 FORMAT(A1)
      IF(CHAR.NE.1HY) GOTO 1
C TEST ENRP
      IF(ENRP.LT.0.0.AND.IERROR.EQ.0) IERROR = -11001
C TEST CORRECTNESS RTEMO
      IF(RTEMO+999.0) 301,301,303
303  IF(RTEMO-240.0) 311,305,305
305  IF(RTEMO-1250.0) 301,301,311
311  IF(IERROR.EQ.0) IERROR = -11002
301  CONTINUE
      RETURN
      END
```

SUBROUTINE VBVK12

C ROUTINE FOR SETTINGS OF VALVES AND IGNITION

COMMON/GEN/NIN(64),NVAL(128),DI(24),DO(4),ACTI,

\$ IERROR,IFLAG1,IFLAG2,IFLAG3,IFLAG4

COMMON/BB1/NTEXP,NUM,AMOX,AM(7),TBURN,APREL,NCOOL

COMMON/BB2/TIGN,TPRE,RTEMO,ENRP,NDASY

COMMON/BB3/NTIM(50),NDIO(50),TI(10),NSIZE,NFL1

COMMON/BB4/LN(7),LM(7),NTEST(24)

C COMMON/BD5/NS(5),NTH(12),NTSCI(14),NTSH(22),NTSHI(32),

C \$ NTSECI(16),NT(5),NVH(12),NVSCI(14),NVSH(22),

C \$ NVSHI(32),NVSECI(16),TVL1,TVL2,TVL3,TVL4

C \$,TVL5

COMMON/BD5/NS(5),NTH(12),NTSCI(12),NTSH(22),NTSHI(32),

\$ NTSECI(16),NT(5),NVH(12),NVSCI(12),NVSH(22),

\$ NVSHI(32),NVSECI(16),TVL1,TVL2,TVL3,TVL4

\$,TVL5

COMMON/DD1/ELAP,DMAS(7),STORE(290)

DIMENSION LI(7),W1(4),W2(4)

DATA LI/97,100,101,102,103,105,107/

C INITIALISE

NSIZE = 0

C TIM CONTAINS TIMING; DIO CONTAINS VALVE,ETC.. INFORMATION

C CHECK WATERCOOLING

I = 0

IF(NCOOL.NE.1) GOTO 50

I = I + 1

NTIM(I) = 0

NDIO(I) = 99

50 CONTINUE

C SET UP TIMING

TI(1) = 0.0

TI(2) = TI(1) + TVL1

C SELECT TYPE OF EXPERIMENT

GOTO(100,101,102,103,104,102,103,107),NTEXP

C H TEST

100 CONTINUE

TI(3) = TI(2) + TVL2

TI(4) = TI(3) + TIGN

TI(5) = TI(4) + TBURN

TI(6) = TI(5) + TVL3

K = 1

DO 200 J=1,NS(K)

NTIM(J+I) = NTH(J)

NDIO(J+I) = NVH(J)

200 CONTINUE

I = I + NS(K)

GOTO 900

C SCI TEST

101 CONTINUE

TI(3) = TI(2) + TPRE

TI(4) = TI(2) + TIGN

TI(5) = TI(4) + TBURN

TI(6) = TI(5) + TVL3

K = 2

DO 201 J=1,NS(K)

NTIM(J+I) = NTSCI(J)

NDIO(J+I) = NVSCI(J)

201 CONTINUE

I = I + NS(K)

GOTO 900

C SH TEST

102 CONTINUE

TI(3) = TI(2) + TVL2

TI(4) = TI(3) + TVL2

TI(5) = TI(4) + TVL5

TI(6) = TI(5) + TBURN

TI(7) = TI(6) + TVL4

TI(8) = TI(6) + TVL3

K = 3

DO 202 J=1,NS(K)

NTIM(J+I) = NTSH(J)

```
NDIO(J+I) = NVSH(J)
202 CONTINUE
  I = I + NS(K)
  GOTO 900
C SHI TEST
103 CONTINUE
  TI(3) = TI(2) + TVL2
  TI(4) = TI(3) + TVL2
  TI(5) = TI(4) + TVL5
  TI(6) = TI(5) + TVL2 * TIGN
  TI(7) = TI(5) + TIGN
  TI(8) = TI(6) + TBURN
  TI(9) = TI(8) + TVL4
  TI(10) = TI(9) + TVL3
  K = 4
  DO 203 J=1,NS(K)
    NTIM(J+I) = NTSHI(J)
    NDIO(J+I) = NVSHI(J)
203 CONTINUE
  I = I + NS(K)
  GOTO 900
C SECI TEST
104 CONTINUE
  TI(3) = TI(2) + TPRE
  TI(4) = TI(3) + TVL2 * TIGN
  TI(5) = TI(3) + TIGN
  TI(6) = TI(4) + TBURN
  TI(7) = TI(6) + TVL4
  TI(8) = TI(6) + TVL3
  K = 5
  DO 204 J=1,NS(K)
    NTIM(J+I) = NTSECI(J)
    NDIO(J+I) = NVSECI(J)
204 CONTINUE
  I = I + NS(K)
  GOTO 900
C NSIZE IS NUMBER OF OPERATIONS DURING TEST
```

```
900 IF(NCOOL.NE.1) GOTO 910
    I = I + 1
    NTIM(I) = NTIM(I-1)
    NDIO(I) = -99
910 NSIZE = I
C GIVE A PRINT-OUT OF THE VALVE SETTINGS
    TYPE 2000
2000 FORMAT(1H0,'VALVE SETTINGS FOR THIS TEST',/,1X,28(1H-),/,
$      1H0,10H  TIME  ,1X,10H  VALVE  ,1X,
$      10HACTIVATION,/,
$      1X,10H  S    ,/)
    DO 300 I=1,NSIZE
    WORD = 'ON '
    IF(NDIO(I).LT.0) WORD = 'OFF'
C IGNITION VITIATOR AND IGNITION SFCC
    IF(IABS(NDIO(I)).EQ.118) GOTO 301
    IF(IABS(NDIO(I)).EQ.119) GOTO 302
    TYPE 2010,TI(NTIM(I)+1),IABS(NDIO(I))-96,WORD
2010 FORMAT(1X,F7.3,7X,I3,8X,A3)
    GOTO 300
    301 TYPE 2011,TI(NTIM(I)+1),WORD
2011 FORMAT(1X,F7.3,1X,9HSP. VITI.,8X,A3)
    GOTO 300
    302 TYPE 2012,TI(NTIM(I)+1),WORD
2012 FORMAT(1X,F7.3,1X,9HSP. SFCC,8X,A3)
300 CONTINUE
C CHECK TIMING
    DO 310 I=2,NT(K)
    IF(TI(I-1).GT.TI(I)) GOTO 999
310 CONTINUE
C GIVE ELAP SUCH A VALUE THAT DURING THE TEST THE STORE
C ARRAY IS EFFICIENT USED.
    ELAP = AINT(TI(NTIM(NSIZE)+1)/10.0) + 1.0
    TYPE 2020,ELAP
2020 FORMAT(1H0,'DATA-STORAGE FOR PRINT-OUT EVERY ',
$      F5.2,' S')
107 CONTINUE
```

```
C PREPARE MANUAL VALVE OPERATION
C ZERO LN AND LM
  DO 350 I =1,7
    LM(I) = 0
    LN(I) = 0
  350 CONTINUE
C COMPARE LINENUMBER AND NDIO
  DO 400 I=1,NSIZE
    DO 410 J=1,7
      IF(IABS(NDIO(I)).EQ.LI(J)) GOTO 415
    410 CONTINUE
      GOTO, 400
    415 LN(J) = 1
      LM(J) = 1
    400 CONTINUE
C LINE 4 IS LINE 3 + VALVE 11
  IF(LN(4).EQ.1) LN(3) = 0
C LINE 6 AND 7 ARE IDENTICAL
  IF(LN(7).EQ.1) LN(6) = 1
C ZERO NFL1, FOR MEANING SEE VBVK13
  NFL1 = 0
C CALL FOR VBVK13 TO DO THE VALVE SETTING
C ZERO NTEST
  DO 402 I=1,24
    NTEST(I) = 0
  402 CONTINUE
  DO 420 I=1,6
    IF(LN(I).EQ.0) GOTO 420
    J = I
    CALL VBVK13(J)
    CALL VBVK17
    LN(I) = J
  420 CONTINUE
C CLOSE VALVES OF LINES NOT IN USE
  DO 500 I=1,22
    IF(NVAL(I).NE.0) GOTO 500
  500 CONTINUE
```

C CHECK NVAL AND NIN FOR CLOSED VALVES

IF(NIN(I).EQ.0) GOTO 500

C CLOSE VALVE I

TYPE 1000,I

1000 FORMAT(1H0,'CLOSE VALVE ',I3)

PAUSE

C CHECK AGAIN

GOTO 520

500 CONTINUE

RETURN

C ERROR CONDITION

999 CONTINUE

IF(IERROR.EQ.0) IERROR = -12001

END

SUBROUTINE VBVK13(IN)

COMMON/GEN/NIN(64),NVAL(128),DI(24),DO(4),ACTI,

\$ IERROR,IFLAG1,IFLAG2,IFLAG3,IFLAG4

COMMON/BB3/ NTIM(50),NDIO(50),TI(10),NSIZE,NFL1

COMMON/BB4/ LN(7),LM(7),NTEST(24)

COMMON/BD6/NST(6),NV(100)

C ROUTINE FOR MANUAL VALVE OPERATION

DIMENSION W1(6),W2(6)

DATA W1/4HAIR ,4HO2(1,4HO2(2,4HO2(3,4HH2 ,4HCH4 /

DATA W2/1H ,3*1H),2*1H /

C NFL1 = 0 NO SETTING OF VALVES DURING THIS ROUTINE

C NFL1 > 0 SETTING OF VALVES DURING THIS ROUTINE

C NTEST CONTAINS INFO ABOUT VALVES FOR THIS LINE

C NVAL CONTAINS VALVE INFORMATION FOR THIS TEST

C IN IS LINE NUMBER

NO = NST(IN)

C JUMP OVER PRINT STATEMENT FOR NFL1 = 0

IF(NFL1.EQ.0) GOTO 7

TYPE 1500,W1(IN),W2(IN)

1500 FORMAT(1H0,'MANUAL SETTING FOR THE ',A4,A1,' LINE',

```

$      /,1X ,33(1H-))
C GET LENGTH
  7 N1 = NV(NO)
C NO IS NV POINTER, N1 IS NUMBER OF DATA LEFT FOR THIS LINE,
C N2 IS ACTION POINTER (E.G.900 OR 999)
  N3 = 0
  1 NO = NO + (N3 + 1)
  N1 = N1 - (N3 + 1)
  IF(N1.LE.0) GOTO 900
  N2 = NV(NO)
C N3 IS NUMBER DATA PER ACTION POINTER
  I = 0
  N3 = 0
 10 I = I + 1
  IF(I.GT.N1) GOTO 15
  IF(IABS(NV(I+NO)).GE.900) GOTO 15
  N3 = N3 + 1
  GOTO 10
 15 CONTINUE
C ASK FOR ACTION
  DO 20 I=1,N3
C READ IO LINE
 22 CONTINUE
  NBIT = NIN(IABS(NV(I+NO)))
  IF(N2-900) 999,31,32
C NECESSARY ACTION
 32 IF(NBIT-1) 40,41,999
C CLOSE VALVE IF OPEN
 41 CONTINUE
  NVAL(IABS(NV(I+NO))) = ISIGN(1,NV(I+NO))
  NTEST(IABS(NV(I+NO))) = ISIGN(1,NV(I+NO))
  IF(NFL1.EQ.0) GOTO 20
  IF(NV(I+NO).GT.0) GOTO 20
  TYPE 1000,IABS(NV(I+NO))
1000 FORMAT(1H0,'CLOSE VALVE ',I3)
  PAUSE
  GOTO 22
```


C OPEN VALVE IF CLOSED

40 CONTINUE

NVAL(IABS(NV(I+NO))) = ISIGN(1,NV(I+NO))

NTEST(IABS(NV(I+NO))) = ISIGN(1,NV(I+NO))

IF(NFL1.EQ.0) GOTO 20

IF(NV(I+NO).LT.0) GOTO 20

TYPE 1010,IABS(NV(I+NO))

1010 FORMAT(1H0,'OPEN VALVE ',I3)

PAUSE

GOTO 22

C SELECT VALVES (OWN CHOICE)

31 IF(NBIT) 20,36,37

36 CONTINUE

NVAL(IABS(NV(I+NO))) = -2

NTEST(IABS(NV(I+NO))) = -2

IF(NFL1.EQ.0) GOTO 20

TYPE 1020,IABS(NV(I+NO))

1020 FORMAT(1H0,'VALVE ',I3,' IS CLOSED',/,

\$ 1X, 'OPEN VALVE IF NECESSARY')

PAUSE

GOTO 20

37 CONTINUE

NVAL(IABS(NV(I+NO))) = +2

NTEST(IABS(NV(I+NO))) = +2

IF(NFL1.EQ.0) GOTO 20

TYPE 1030,IABS(NV(I+NO))

1030 FORMAT(1H0,'VALVE ',I3,' IS OPEN',/,

\$ 1X, 'CLOSE VALVE IF NECESSARY')

PAUSE

20 CONTINUE

GOTO 1

900 CONTINUE

C CHECK NTEST AGAINST NIN

NBIT = 0

DO 910 I=1,24

IF(IABS(NTEST(I)).GT.1) GOTO 910

IF(NTEST(I)) 912,910,914

```
912 IF(NIN(I).NE.0) NBIT = 1
      GOTO 910
914 IF(NIN(I).NE.1) NBIT = 1
910 CONTINUE
      IF(NBIT.EQ.0) IN = 0
      RETURN
999 IF(IERROR.EQ.0) IERROR = -13001
      RETURN
      END
```

SUBROUTINE VBVK14

```
C ROUTINE TO ADJUST AND CHECK THE MASS FLOWS
COMMON/GEN/NIN(64),NVAL(128),DI(24),DO(4),ACTI,
$      IERROR,IFLAG1,IFLAG2,IFLAG3,IFLAG4
COMMON/BB1/NTEXP,NUM,AMOX,AM(7),TBURN,APREL,NCOOL
COMMON/BB2/TIGN,TPRE,RTEMO,ENRP,NDASY
COMMON/BB3/NTIM(50),NDIO(50),TI(10),NSIZE,NFL1
COMMON/BB4/LN(7),LM(7),NTEST(24)
COMMON/BD1/TO(4),CO(4),C1(4),C2(4),C3(4),C4(4)
COMMON/BD2/GM(7),V(18),VL(7),NPB(18),NBB(7)
COMMON/BD3/AMARK1(7),AMARK2(7),NPP(7),NPS(7),NTT(7)
COMMON/BD4/TNUL(7),PL(3,7),Z1(7),Z2(7),Z3(7),Y1(7),Y2(7)
COMMON/BD7/FLM1,FLM2,FLM3,FLM4,FLM5,FLM6,FLM7,IJPM(7)
DIMENSION NE(7),NZ(7),NQ(8),NV(7)
C NE CONTAINS NUMBER OF TIMES AN ERROR HAS OCCURRED FOR
C THE PARTICULAR LINE (I.E. P,S,* OR R MARK)
C NZ CONTAINS BIT NUMBERS OF ELECTRICAL VALVES TO OPEN
C DURING LOWERING SECONDARY PRESSURES
      DATA NZ/97,100,101,102,104,106,108/
C NQ CONTAINS BIT NUMBERS OF READY FOR *** LAMPS
      DATA NQ/65,66,66,66,67,68,68,999/
C NV IS USED TO CALCULATE OPERATION TIMES
      DATA NV/97,100,101,102,103,105,107/
C CALCULATE MASS FLOWS
      CALL VBVK15
```

```
IF(IERROR.NE.0) RETURN
C OPEN ELECTRICAL VALVES FOR PRESSURE MEASUREMENTS
C HYDROGEN
    IF(LM(5).NE.0) NVAL(103) = 1
C METHANE
    IF(LM(6).NE.0) NVAL(105) = 1
    IF(LM(7).NE.0) NVAL(107) = 1
C INITILISE
    DO 20 K=1,2
C GIVE COMMENT
    IF(K.EQ.1) TYPE 2000
    IF(K.EQ.2) TYPE 2100
2000 FORMAT(1H0,'THE FOLLOWING INFORMATION IS USERS INFORMATION',
$          ' ONLY, CHECK AND CORRECT IF POSSIBLE',/,
$          1X,82(1H-))
2100 FORMAT(1H0,'THE FOLLOWING INFORMATION WILL BE USED TO',
$          ' SET UP THE CORRECT MASSFLOWS',/,1X,70(1H-))
    DO 15 I=1,7
C ZERO ERROR POINTER
    NE(I) = 0
    IF(LM(I).EQ.0) GOTO 15
C ASK FOR USED DIAFRAGM,PENETRATION LENGTH OR NUMBER OF TURNS
    1 TYPE 1000
1000 FORMAT(/,1H$, 'GIVE THE USED ')
C JUMP TO THE REQUIRED LINE
    CALL VBVK19(I)
    TYPE 1010,AMARK1(I),AMARK2(I)
1010 FORMAT(1H$, ' FOR THE ',2A3,' LINE ? * ')
    ACCEPT *,PL(1,I)
    TYPE 1020
1020 FORMAT(1X)
    CALL VBVK19(I)
    TYPE 1030,PL(1,I)
1030 FORMAT(1H$, ' IS ',F5.2)
    TYPE 1020
    TYPE 1040
1040 FORMAT(/,1H$, 'CORRECT (Y OR N) ? * ')
```

```
ACCEPT 999,CHAR
999: FORMAT(A1)
C CHECK CORRECT INPUT
  A1 = AMIN1(PL(2,I),PL(3,I))
  A2 = AMAX1(PL(2,I),PL(3,I))
  IF(CHAR.NE.1HY.OR.PL(1,I).LT.A1.
$           OR.PL(1,I).GT.A2) GOTO 1
15 CONTINUE
  N3 = 0
  DO 10 J=1,7
C IGNORE LINES NOT IN USE
  IF(LM(J).EQ.0) GOTO 200
  N3 = 1
C SCAN OVER OPERATIONS TO OBTAIN OPERATION-TIME OF THIS LINE
  N1 = 0
  N2 = 0
  DO 100 I=1,NSIZE
  IF(NDIO(I).EQ.NV(J)) N1 = NTIM(I)
  IF(NDIO(I).EQ.-NV(J)) N2 = NTIM(I)
100 CONTINUE
  T = TI(N2+1) - TI(N1+1)
C CALCULATE SECONDARY PRESSURE LIMITS FOR THIS MASSFLOW
  CALL VBVK16(J,P1,P2)
  IF(IERROR.NE.0) RETURN
  N1 = 0
C USE N2 TO OBTAIN THE USED BUFFER VOLUMES
  N2 = 0
  IF(J.EQ.1) GOTO 120
  DO 110 I=1,J-1
  N2 = N2 + NBB(I)
110 CONTINUE
C ASK FOR CORRECTNESS MASSFLOW WHEN MORE THAN 5 TIMES
C AN ERROR HAS BEEN DETECTED
120 IF(K.LT.2.OR.NE(J).LE.5) GOTO 125
  TYPE 1050,AMARK1(J),AMARK2(J)
1050 FORMAT(/,1H$, 'DO YOU WANT TO CHANGE THE MASS',
$           'FLOW OF THE ',2A3,' LINE (Y OR N) ? * ')
```

```
ACCEPT 999,CHAR
IF(CHAR.NE.1HY) GOTO 125
C ASK NEW MASSFLOW
122 TYPE 1060,AMARK1(J),AMARK2(J),AM(J)
1060 FORMAT(/,1H$, 'MASSFLOW ',2A3,' IS ',F10.4,
$          ' KG/S, GIVE NEW VALUE ? * ')
ACCEPT *,DSW
TYPE 1070,DSW
1070 FORMAT(/,1H$, 'NEW MASSFLOW IS ',F10.4,' KG/S',
$          ' CORRECT (Y OR N) ? * ')
ACCEPT 999,CHAR
IF(CHAR.NE.1HY) GOTO 122
AM(J) = DSW
CALL VBVK16(J,P1,P2)
C IS IT OK NOW ?
TYPE 1080
1080 FORMAT(/,1H$, 'IS THE MASSFLOW AT THIS TIME CORRECT',
$          ' (Y OR N) ? * ')
ACCEPT 999,CHAR
IF(CHAR.NE.1HY) GOTO 122
C ZERO ERROR POINTER
NE(J) = 0
125 MKP = ' '
MKS = ' '
C CALCULATE USED BUFFERVOLUME AT THIS TIME
VTOT = VL(J)
DO 130 I=1,NBB(J)
VTOT = VTOT + FLOAT(NIN(NPB(N2+I))) * V(N2+I)
130 CONTINUE
C CALCULATE DEL(P) FOR THIS LINE
DQ = AM(J) * T * 8314.34 * DI(NTT(J)) / (GM(J)*VTOT)
C CHECK PRIMARY PRESSURE AGAINST CALCULATED SECONDARY PRESSURE
C CHECK ON SECONDARY PRESSURE RANGE FOR K=1 AND ON THE
C REAL SECONDARY PRESSURE FOR K=2 (REAL INSTALLATION)
IF(K.EQ.1.AND.
$  DQ+P1.GT.0.95*DI(NPP(J))) MKP = 'P'
IF(K.EQ.2.AND.
```

```
$      DQ+DI(NPS(J)).GT.0.95*DI(NPP(J)) MKP = 'P'
C CHECK SEC.PRES. AGAINST CALCULATED SEC.PRES.
      IF(DI(NPS(J)).LT.P1.OR.DI(NPS(J)).GT.P2) MKS = 'S'
C CHECK CALC.SEC.PRES.AGAINST 2*APREL
      IF(K.EQ.1.AND.FLM7*APREL.GT.P1 ) MKS = '*'
      IF(K.EQ.2.AND.
$      FLM7*APREL.GT.DI(NPS(J))) MKS = 'S'
C CHECK P1 AND P2 FOR CORRECT INTERVAL
      IF(P1.GE.P2) MKS = '*'
C MKP = P MEANS PRIM.PRESSURE TOO LOW
C MKS = S MEANS SEC.PRESSURE OUT OF RANGE
C      OR LESS THAN TWICE THE SFCC PRESSURE
C MKS = * MEANS NO TEST POSSIBLE UNDER THIS CONDITIONS
C CALCULATE MASSFLOW FOR THESE SETTINGS
      DSW = (Z1(J)*PL(1,J)*PL(1,J)+Z2(J)*PL(1,J)+Z3(J)) *
$      (Y1(J)*DI(NPS(J))+Y2(J))
C CORRECT FOR REF TEMPERATURE FOR SCMC
      IF(TNUL(J).NE.0.0) DSW = DSW *
$      SQRT(TNUL(J)/DI(NTT(J)))
C INFO ONLY FOR K=1
      TYPE 1100,AMARK1(J),AMARK2(J),T,VTOT*1.0E+03,
$      DQ*1.0E-05,DI(NPP(J))*1.0E-05,DI(NPS(J))*1.0E-05,
$      P1*1.0E-05,P2*1.0E-05,DSW,MKP,MKS,FLM7,FLM7
1100 FORMAT(1H0,'LINE ',2X,'OPERATION TIME',2X,
$      'BUFFERVOLUME',2X,'DEL(P)',2X,
$      'PRIM.PRES.',2X,'SEC. PRES.',2X,
$      'SEC.PRES.RANGE',2X,'MASSFLOW',2X,
$      'MKP  MKS',/,
$      1H , ' ',2X,' S. ',2X,
$      ' L ',2X,' BAR ',2X,
$      ' BAR ',2X,' BAR ',2X,
$      ' BAR ',2X,' KG/S',//,
$      1H ,2A3,6X,F6.2,6X,2X,F10.3,2X,F6.2,2X,
$      F10.3,2X,F10.3,2X,F6.2,'--',F6.2,2X,
$      F9.4,4X,A1,5X,A1,/,
$      ' MKP = P MEANS PRIMARY PRESSURE TOO LOW',/,
$      ' MKS = S MEANS SECONDARY PRESSURE OUT OF RANGE',
```

```
$          ' OR LESS THAN ',F4.2,' TIMES THE SFCC PRESSURE',/,
$          ' MKS = * MEANS NO TEST POSSIBLE FOR THESE',
$          ' CONDITIONS, I.E. ',F4.2,' TIMES THE SFCC ',
$          'PRESSURE',
$          /,'          EXCEEDS THE SECONDARY PRESSURE',
$          ' OR CALCULATED PRESSURE RANGE',
$          ' IS NOT REALISTIC')
      IF(K.LT.2) GOTO 10
C ACTION FOR K=2
      N3 = 1
C CHECK POSSIBILITY OF THIS TEST
      IF(MKS.NE.'*') GOTO 140
C INCREASE ERROR POINTER
      NE(J) = NE(J) + 10
      TYPE 1110,DI(NPS(J))*1.0E-05,FLM7,APREL*1.0E-05,P1*1.0E-05,
$          P2*1.0E-05
1110 FORMAT(1H0,'TEST IS NOT POSSIBLE UNDER THESE CONDITIONS',
$          ', BECAUSE THE SECONDARY PRESSURE OF ',
$          'F6.2,' BAR IS',/, ' EITHER BELOW ',
$          'F4.2,' TIMES',
$          ' THE SFCC PRESSURE OF ',F6.2,' BAR OR/AND',
$          ' THE CALCULATED PRESSURE RANGE',/,
$          '1X,'OF ',F6.2,' - ',F6.2,' BAR IS NOT REALISTIC')
C RETURN TO LABEL 120 IN CASE OF NON-REALISTIC
      IF(P1.GE.P2) GOTO 120
C CHECK PRIMARY PRESSURE
      140 IF(MKP.NE.'P') GOTO 150
C INCREASE ERROR POINTER
      NE(J) = NE(J) + 1
C PRIMARY PRESSURE TOO LOW
      TYPE 1120,AMARK1(J),AMARK2(J)
1120 FORMAT(1H0,'THE PRIMARY PRESSURE OF THE ',2A3,' LINE ',
$          'IS TOO LOW')
C POSSIBLE INTERRUPT BY OPERATOR
      TYPE 1500
1500 FORMAT(/,1H$, 'GIVE STOP/RETURN TO STOP/CONTINUE * ')
      ACCEPT 998,CHAR
```

```
998 FORMAT(A4)
    IF(CHAR.NE.4H .AND.IERROR.EQ.0) IERROR = -14001
    IF(IERROR.NE.0) RETURN
C FOR SECOND TIME CALL VBVK13
    N1 = N1 + 1
    IF(N1.GT.1) GOTO 180
C CHECK IF ALL BUFFERVESSELS ARE OPEN
    DO 160 I=1,NBB(J)
    IF(NIN(NPB(N2+I)).EQ.0) GOTO 170
160 CONTINUE
C ALL BUFFERS ARE OPEN, CORRECT PRIM.PRES. BY CALLING VBVK13
180 I = J
C SET NFL1 TO 1 TO OBTAIN SOME ACTION
    NFL1 = 1
    CALL VBVK13(I)
    CALL VBVK17
C RESET NFL1 AFTER ACTION
    NFL1 = 0
    N1 = 0
    GOTO 120
C NOT ALL BUFFERS ARE OPEN, TRY TO CORRECT PRIM.PRES.BY
C OPENING MORE BUFFERS
170 TYPE 1130
1130 FORMAT(1H0,'INCREASE IF POSSIBLE, PRIMARY PRESSURE',
    $          ' BY OPENING : ',/)
    DO 190 I=1,NBB(J)
    IF(NIN(NPB(N2+I)).EQ.0) TYPE 1140,NPB(N2+I)
C SET NTEST TO 2 TO INDICATE POSSIBLE USE OF BUFFER VESSEL
C SEE ALSO VBVK17
    NTEST(NPB(N2+I)) = 2
1140 FORMAT(1H , 'VALVE ',I3)
190 CONTINUE
    PAUSE
C CHECK BUFFERVESSELS WITH VBVK17
    CALL VBVK17
C POSSIBLE INTERRUPT BY OPERATOR
    TYPE 1500
```



```
ACCEPT 998,CHAR
IF(CHAR.NE.4H .AND.IERROR.EQ.0) IERROR = -14001
IF(IERROR.NE.0) RETURN
GOTO 120

C PRIMARY PRESSURE OK, TEST SEC.PRESSURE
150 IF(MKS.EQ.' ') GOTO 200
C INCREASE ERROR POINTER
NE(J) = NE(J) + 1
TYPE 1150,AMARK1(J),AMARK2(J),DI(NPS(J))*1.0E-05
1150 FORMAT(1H0,'THE SECONDARY PRESSURE OF THE ',2A3,
$ ' LINE IS ',F6.2,' BAR')
C CHECK SFCC PRESSURE
IF(FLM7*APREL.GT.DI(NPS(J))) GOTO 215
TYPE 1155,P1*1.0E-05,P2*1.0E-05
1155 FORMAT(1X,'AND LIES NOT BETWEEN ',F6.2,' BAR AND ',
$ F6.2,' BAR')
IF(DI(NPS(J)).GT.P1) GOTO 210
C SEC.PRES. TOO LOW
215 TYPE 1160
1160 FORMAT(1H0,'THE SECONDARY PRESSURE IS TOO LOW, ',
$ ' INCREASE PRESSURE')
PAUSE
C POSSIBLE INTERRUPT BY OPERATOR
TYPE 1500
ACCEPT 998,CHAR
IF(CHAR.NE.4H .AND.IERROR.EQ.0) IERROR = -14001
IF(IERROR.NE.0) RETURN
GOTO 120
210 IF(DI(NPS(J)).LT.P2) GOTO 200
TYPE 1170
1170 FORMAT(1H0,'THE SECONDARY PRESSURE IS TOO HIGH')
C LOWER PRESSURE BY OPENING ELECTRICAL VALVE
TYPE 1180,AMARK1(J),AMARK2(J)
1180 FORMAT(1H0,'CLOSE REDUCE VALVE OF THE ',2A3,
$ ' LINE, GIVE A RETURN AFTER CLOSING')
PAUSE
C POSSIBLE INTERRUPT BY OPERATOR
```

```
TYPE 1500
ACCEPT 998,CHAR
IF(CHAR.NE.4H .AND.IERROR.EQ.0) IERROR = -14001
IF(IERROR.NE.0) RETURN
TYPE 1190,NZ(J)-96
1190 FORMAT(1H0,'THE SECONDARY PRESSURE WILL BE LOWERED ',
$          'BY OPENING VALVE ',I3)
NVAL(NZ(J)) = 1
C SET IFLAG4 TO 1, ONLY CALLS TO VBVK30 IN VBVK14
IFLAG4 = 1
230 CALL VBVK30(1)
CALL VBVK30(2)
CALL VBVK30(3)
CALL VBVK30(4)
IF(IERROR.NE.0) RETURN
IF(DI(NPS(J)).GT.1.0E+05) GOTO 230
C SEC.PRESSURE REDUCED TO LESS THAN 1 BAR
NVAL(NZ(J)) = 0
CALL VBVK30(2)
C RESET IFLAG4 TO 0
IFLAG4 = 0
C INCREASE SECONDARY PRESSURE AGAIN TO CORRECT VALUE
TYPE 1195
1195 FORMAT(1H0,'INCREASE SECONDARY PRESSURE TO CORRECT',
$          ' VALUE')
PAUSE
GOTO 120
C SEC.PRESSURE OK, CALCULATE SETTING FOR THIS LINE
C PUT ON READY FOR ***
200 IF(NQ(J+1).EQ.NQ(J).OR.N3.EQ.0) GOTO 10
IF(K.EQ.2) NVAL(NQ(J)) = 1
N3 = 0
10 CONTINUE
TYPE 1200
1200 FORMAT(1H0,'CHECK PRIMARY AND SECONDARY PRESSURES',
$          /,1H0,'CHECK AND CORRECT IF NECESSARY THE MASS',
$          'FLOW DEVICES')
```

```
PAUSE
IF(K.GT.1) GOTO 20
C CALL VBVK13 AND VBVK17 TO CHECK AGAIN THE MANUAL VALVES
C BUT FIRST RESET NFL1
NFL1 = 0
DO 25 I=1,6
IF(LM(I).EQ.0) GOTO 25
JJ = I
CALL VBVK13(JJ)
IF(IERROR.NE.0) RETURN
LN(I) = JJ
25 CONTINUE
C SET NFL1 TO 1
NFL1 = 1
C DO MANUAL SETTING FOR LINES WITH NON CORRECT VALVE SETTING
C LINE 3 AND 4 ARE IDENTICAL EXCEPT VALVE 11
IF(LN(4).NE.0) LN(3) = 0
DO 30 I=1,6
IF(LN(I).EQ.0) GOTO 30
II = I
CALL VBVK13(II)
IF(IERROR.NE.0) RETURN
30 CONTINUE
CALL VBVK17
IF(IERROR.NE.0) RETURN
20 CONTINUE
DO 115 I=1,7
IF(LM(I).EQ.0) GOTO 115
C ASK FOR USED DIAFRAGM,PENETRATION LENGTH OR NUMBER OF TURNS
101 TYPE 1000
C JUMP TO THE REQUIRED LINE
CALL VBVK19(I)
TYPE 1010,AMARK1(I),AMARK2(I)
ACCEPT *,PL(1,I)
DSW = (Z1(I)*PL(1,I)*PL(1,I)+Z2(I)*PL(1,I)+Z3(I)) *
$ (Y1(I)*DI(NPS(I))+Y2(I))
C CORRECT FOR REF TEMPERATURE FOR SCMC
```

```
IF(TNUL(I).NE.0.0) DSW = DSW *
$           SQRT(TNUL(I)/DI(NTT(I)))
TYPE 1020
CALL VBVK19(I)
TYPE 1030,PL(1,I)
C GIVE A PRINT-OUT OF THE MASS-FLOW WITH THIS SETTING
TYPE 1032,DSW
1032 FORMAT(1H$, ' ,WHICH GIVES A MASS-FLOW OF ',F7.4,' KG/S')
TYPE 1020
TYPE 1040
ACCEPT 998,CHAR
C CHECK CORRECT INPUT
A1 = AMIN1(PL(2,I),PL(3,I))
A2 = AMAX1(PL(2,I),PL(3,I))
IF(CHAR.NE.1HY.OR.PL(1,I).LT.A1.
$           OR.PL(1,I).GT.A2) GOTO 101
115 CONTINUE
C GIVE A PRINT-OUT
TYPE 1210
1210 FORMAT(1H0,'SETTINGS FOR USED LINES',/,1X,23(1H-),/,
$           1H0,'MASSFLOW IN KG/S AND SECONDARY PRESSURE',
$           ' IN PASCAL')
DO 116 I=1,7
IF(LM(I).EQ.0) GOTO 116
TYPE 1220,AMARK1(I),AMARK2(I),Z1(I),PL(1,I),Z2(I),
$           PL(1,I),Z3(I),Y1(I),Y2(I)
1220 FORMAT(1H0,'MASSFLOW (',2A3,') = (',E12.5,' * (',
$           E12.5,' ) ** 2 + ',E12.5,' * ',E12.5,
$           ' + ',/,22X,' + ',E12.5,
$           ' ) * ( ',E12.5,' * SEC.PRESSURE + ',
$           E12.5,' )')
TYPE 1230
1230 FORMAT(/,1H$, 'TO GET THE CORRECT MASSFLOW AT THE',
$           ' ACTUAL TEMPERATURE, WE HAVE TO MULTIPLY',
$           ' THE MASSFLOW WITH ')
IF(TNUL(I).EQ.0.0) GOTO 117
TYPE 1020
```

```
TYPE 1250, TNUL(I)
1250 FORMAT(1X, 'THE SQRT ( ', F6.2, ' / TEMPERATURE OF THE GAS )')
GOTO 116
117 TYPE 1240
1240 FORMAT(' 1.00')
TYPE 1020
116 CONTINUE
C POSSIBLE INTERRUPT BY OPERATOR
IF(IERROR.NE.0) RETURN
C CHECK NVAL AGAINST NIN
DO 40 I=1,24
IF(NVAL(I).EQ.NIN(I)) GOTO 40
C THIS VALVE IS NOT CORRECT
TYPE 1201, I, NVAL(I), NIN(I)
1201 FORMAT(1H0, '*****WARNING : SETTING OF VALVE ', I2,
$ ' HAS CHANGED',
$' (INPUTBIT HAS CHANGED FROM ', I2, ' TO ', I2, ')', //,
$1X, 'CHECK THIS NEW SETTING AND CHANGE IF WRONG')
PAUSE
C ACCEPT THE FOUND INPUTBIT
NVAL(I) = NIN(I)
TYPE 1202, NVAL(I), I
1202 FORMAT(1H0, '*****WARNING : INPUTBIT = ', I2, ' ACCEPTED',
$ ' FOR VALVE ', I2)
40 CONTINUE
C SET IFLAG1 FOR CHECKING OF VALVES BY VBVK01
IFLAG1 = 1
RETURN
END
```

SUBROUTINE VBVK15

C ROUTINE FOR CALCULATING ALL MASS FLOWS

COMMON/GEN/NIN(64),NVAL(128),DI(24),DO(4),ACTI,

\$ IERROR,IFLAG1,IFLAG2,IFLAG3,IFLAG4

COMMON/BB1/NTEXP,NUM,AMOX,AM(7),TBURN,APREL,NCOOL

COMMON/BB2/TIGN,TPRE,RTEMO,ENRP,NDASY

COMMON/BB3/NTIM(50),NDIO(50),TI(10),NSIZE,NFL1

COMMON/BD1/TO(4),CO(4),C1(4),C2(4),C3(4),C4(4)

COMMON/BD3/AMARK1(7),AMARK2(7),NPP(7),NPS(7),NTT(7)

COMMON/BD7/FLM1,FLM2,FLM3,FLM4,FLM5,FLM6,FLM7,IJPM(7)

DIMENSION DJ(2)

C CALCULATE MASS FLOWS

C LINE 1..7 REPRESENT M(AIR),M(O2(1)),M(O2(2)),M(O2(3)),

C M(H2),M(CH4(1)) AND M(CH4(2)) RESP.

1 AM(1) = 0.0

AM(2) = 1.0

AM(6) = 0.0

C FOR H TEST NO VITIATOR

IF(NTEXP.EQ.1) GOTO 40

8 CONTINUE

C CALCULATE GAS-TEMPERATURE AS A MEAN

TJ = (DI(3)+DI(9)+DI(17)) / 3.0

IF(TJ.LT.TO(1).OR.TJ.GE.TO(4)) GOTO 7

GOTO 9

7 TYPE 1500,TJ,TO(1),TO(4)

1500 FORMAT(1H0,'TEMPERATURE OUT OF RANGE',/,

\$ 1H0,'THE GAS TEMPERATURE IS ',F6.1,' K ',/,1H0,

\$ 'THIS VALUE LIES NOT BETWEEN ',F6.1,' AND ',

\$ F6.1,' K',/,1H0,'RE-ADJUST',/)

C RETURN IN CASE OF ERROR

IF(IERROR.NE.0) RETURN

PAUSE

C POSSIBLE INTERRUPT BY OPERATOR

TYPE 1505

1505 FORMAT(/,1H\$, 'GIVE STOP/RETURN TO STOP/CONTINUE *')

ACCEPT 999,CHAR

999 FORMAT(A4)

```
IF(CHAR.EQ.4H ) GOTO 8
IF(IERROR.EQ.0) IERROR = -15001
RETURN
9 CONTINUE
C RETURN IN CASE OF ERROR
IF(IERROR.NE.0) RETURN
C ADJUST RTEMO FOR COLD EXPERIMENTS
IF(RTEMO.EQ.-999.0) RTEMO = TJ
N1 = 1
N2 = 2
DO 25 I=1,4
IF(TO(I).GT.TJ) GOTO 30
N1 = I
N2 = I + 1
25 CONTINUE
IF(IERROR.EQ.0) IERROR = -15002
RETURN
30 CONTINUE
IF(I.NE.1) GOTO 33
IF(IERROR.EQ.0) IERROR = -15002
RETURN
33 CONTINUE
A1 = (TJ-TO(N1)) / (TO(N2)-TO(N1))
C INTERPOLATE BETWEEN TABLE AND ACTUAL TEMPERATURE
J = 0
DO 15 I=N1,N2
J = J + 1
DJ(J) = CO(I) + C1(I) * RTEMO + C2(I) * RTEMO * RTEMO
DJ(J) = DJ(J) + C3(I) * RTEMO * RTEMO * RTEMO
DJ(J) = DJ(J) + C4(I) * RTEMO ** 4
15 CONTINUE
AM(6) = DJ(1) + A1 * (DJ(2) - DJ(1))
AM(6) = AM(6) / 1000.0
AM(2) = 5.575 * AM(6)
AM(1) = 1.0 - 6.575 * AM(6)
C CALCULATE REAL MASS FLOWS
40 AM(1) = AM(1) * AMOX * (1.0 - ENRP / 100.0)
```

```
AM(2) = AM(2) * AMOX * (1.0 - ENRP / 100.0)
AM(6) = AM(6) * AMOX * (1.0 - ENRP / 100.0)
C CALCULATE AM(2) ACCORDING TO THE ENRICHMENT
AM(2) = AM(2) + (ENRP/100.0) * AMOX
C CALCULATE MASS FLOWS FOR ALL THE OTHER LINES
GOTO (51,52,53,54,52,53,54,54) NTEXP
C H TEST
C THE FACTOR FLM1 IS FOR HYDROGEN
C THE MAXIMUM FLOW FOR H2 IS FLM2 KG/S >>>>>>>>
51 AM(5) = AMIN1(FLM1*AM(2),FLM2)
GOTO 60
C SCI AND SECI TEST
52 AM(4) = FLM3 * AM(1)
AM(5) = AMIN1(FLM1*AM(4),FLM2)
GOTO 60
C SH AND SEH TEST
53 AM(3) = FLM6
AM(7) = FLM4 * AM(3)
GOTO 60
C SHI AND SEHI TEST
54 AM(3) = FLM6
AM(4) = FLM3 * AM(1)
AM(5) = AMIN1(FLM1*AM(4),FLM2)
AM(7) = FLM4 * AM(3)
60 CONTINUE
TYPE 2000, AMOX, (AMARK1(I), AMARK2(I), AM(I), I=1, 7)
2000 FORMAT(1H0, 'CALCULATED MASS FLOWS', /, 1X, 21(1H-), /,
$ 1H0, 'MASS FLOW OXIDIZER ', F10.4, ' KG/S',
$ 7(/, 1H , 'MASS FLOW ', 2A3, 3X, F10.4, ' KG/S'))
RETURN
END
```



```

SUBROUTINE VBVK16(J,P1,P2)
COMMON/GEN/NIN(64),NVAL(128),DI(24),DO(4),ACTI,
$      IERROR,IFLAG1,IFLAG2,IFLAG3,IFLAG4
COMMON/BB1/NTEXP,NUM,AMOX,AM(7),TBURN,APREL,NCOOL
COMMON/BD3/ AMARK1(7),AMARK2(7),NPP(7),NPS(7),NTT(7)
COMMON/BD4/ TNUL(7),PL(3,7),Z1(7),Z2(7),Z3(7),Y1(7),Y2(7)
COMMON/BD7/FLM1,FLM2,FLM3,FLM4,FLM5,FLM6,FLM7,IJPM(7)
C CALCULATE MASS FLOW AT REF. TEMPERATURE TNUL FOR SCMC
  AMO = AM(J)
  IF(TNUL(J).EQ.0.0) GOTO 102
  AMO = AM(J) * SQRT(DI(NTT(J))/TNUL(J))
C CALCULATE ALL LIMITS ON PRESSURES AND LENGTHS
102 DO 200 I=1,3
  AT(I) = Z1(J)*PL(I,J)*PL(I,J)+Z2(J)*PL(I,J)+Z3(J)
  PW(I) = (AMO / AT(I) - Y2(J)) / Y1(J)
200 CONTINUE
C CALCULATE DIAFRAGM,PENETRATION LENGTH OR NUMBER OF TURNS
C FOR THIS SECONDARY PRESSURE
  ATOBS = AMO / (Y1(J)*DI(NPS(J)) + Y2(J))
  A1 = Z2(J)*Z2(J) - 4.0*Z1(J)*(Z3(J)-ATOBS)
C SET A2 TO NONSENSE
  A2 = -1.0E+30
  IF(A1.LT.0.0) GOTO 210
C IT IS POSSIBLE TO CALCULATE ONE OF THESE PROPERTIES
  A2 = (-Z2(J)-SQRT(A1)) / (2.0*Z1(J))
  A1 = (-Z2(J)+SQRT(A1)) / (2.0*Z1(J))
  GOTO 220
210 A1 = 1.0E+30
220 ATOBS = AMAX1(A1,A2)
C TEST IF THE OTHER SOLUTION IS A REASONABLE ONE
  A1 = AMIN1(A1,A2)
  IF(A1.LT.0.0) A1 = 1.0E+30
C PRINT ALL LIMITS
  TYPE 2000,AMARK1(J),AMARK2(J),AM(J),DI(NTT(J)),
$      DI(NPS(J))*1.0E-05
2000 FORMAT(1HO,'DETAILS FOR THE ',2A3,' LINE ',
$      75(1H),//,
```

```
$      1H , 'REQUIRED MASSFLOW (KG/S) IS', T30, F10.3, /,  
$      1H , 'GASTEMPERATURE (K) IS', T30, F10.3, /,  
$      1H , 'SECONDARY PRESSURE (BAR) IS', T30, F10.3)  
TYPE 1020  
1020 FORMAT(1X)  
CALL VBVK19(J)  
TYPE 2010, PL(1, J)  
2010 FORMAT(1H$, ' IS ', F10.3)  
TYPE 1020  
TYPE 2020  
2020 FORMAT(/, 1H$, 'THE '  
CALL VBVK19(J)  
TYPE 2030, DI(NPS(J))*1.0E-05, ATOBS  
2030 FORMAT(1H$, ' FOR A SECONDARY PRESSURE OF ', F7.2,  
$      ' BAR SHOULD BE ', F7.2)  
C PRINT ALSO THE OTHER SOLUTION FOR A1.NE.INFINITY  
IF(A1.NE.1.0E+30) TYPE 2035, A1  
2035 FORMAT(1H$, ' OR ', F7.2)  
TYPE 1020  
TYPE 2040  
2040 FORMAT(1H$, 'THE SECONDARY PRESSURE FOR A '  
CALL VBVK19(J)  
TYPE 2050, PL(1, J), PW(1)*1.0E-05  
2050 FORMAT(1H$, ' OF ', F7.2, ' SHOULD BE ', F7.2, ' BAR')  
TYPE 1020  
TYPE 2060  
2060 FORMAT(1H0, 'POSSIBLE SECONDARY PRESSURE RANGE', /,  
$      1X, 33(1H-), /)  
TYPE 2070  
2070 FORMAT(1H$, 'FOR A '  
CALL VBVK19(J)  
TYPE 2080, PL(2, J), PW(2)*1.0E-05  
2080 FORMAT(1H$, ' OF ', F7.2, ' WE NEED A SECONDARY PRESSURE',  
$      ' OF ', F7.2, ' BAR')  
TYPE 1020  
TYPE 2070  
CALL VBVK19(J)
```

```
TYPE 2080,PL(3,J),PW(3)*1.0E-05
TYPE 1020
P1 = AMIN1(PW(2),PW(3))
P2 = AMAX1(PW(2),PW(3))
C THE MAXIMUM ALLOWABLE SECONDARY PRESSURE IS 80 BAR
C FLM5 IS THE MAXIMUM ALLOWABLE SECONDARY PRESSURE
P2 = AMIN1(P2,FLM5)
RETURN
END
```

```
SUBROUTINE VBVK17
C ROUTINE FOR OPTIONAL MANUAL VALVES (BUFFER VESSELS)
COMMON/BB3/NTIM(50),NDIO(50),TI(10),NSIZE,NFL1
COMMON/BB4/ LN(7),LM(7),NTEST(24)
COMMON/GEN/NIN(64),NVAL(128),DI(24),DO(4),ACTI,
$          IERROR,IFLAG1,IFLAG2,IFLAG3,IFLAG4
DIMENSION W2(2),W3(2)
BYTE CHAR
DATA W2/3HCLO,3HOPE/
DATA W3/3HSED,3HN /
C GIVE SETTING OF BUFFERS FOR NFL1 = 1 ; ONLY THOSE WHO ARE
C ACTIVATED BY VBVK13 ( LOOK IN NTEST )
IF(NFL1.EQ.0) GOTO 100
10 TYPE 2000
2000 FORMAT(1H0,'BUFFER VESSEL          CONDITION',/,
$          1H,'-----'          '-----',/)
C SCAN NTEST FOR -2 OR +2,INDICATING BUFFER VESSEL
C COUNT THE NUMBER OF VESSELS WITH N1
N1 = 0
DO 20 I=1,24
IF(IABS(NTEST(I)).LE.1) GOTO 20
C PRINT THIS ONE
TYPE 2005,I,W2(NIN(I)+1),W3(NIN(I)+1)
2005 FORMAT(6X,I2,14X,2A3)
N1 = N1 + 1
```

```
20 CONTINUE
C IGNORE CORRECTION OF VESSELS FOR N1 = 0
  IF(N1.NE.0) GOTO 30
  TYPE 2007
2007 FORMAT(1X,'NONE')
  GOTO 100
C ASK FOR CORRECT SETTING
  30 TYPE 2010
2010 FORMAT(/,1H$, 'IS THE SETTING OF THESE BUFFER VESSELS',
  $           ' CORRECT. (Y OR N) ? * ')
  ACCEPT 2015,CHAR
2015 FORMAT(A1)
  IF(CHAR.EQ.1HY) GOTO 100
C CORRECT THESE BUFFER
  TYPE 2020
2020 FORMAT(1H0,'CORRECT THE SETTING OF THESE BUFFER VESSELS')
  PAUSE
  GOTO 10
C CHECK MANUAL SETTING AND GIVE VALUES TO NVAL FOR NFL1 = 1
C ZERO ALSO NTEST
  100 DO 110 I=1,24
    IF(NFL1.EQ.0) GOTO 190
    GOTO (120,130,190,140,120),NTEST(I)+3
  120 NVAL(I) = NIN(I)
    GOTO 150
  130 NVAL(I) = 0
    GOTO 150
  140 NVAL(I) = 1
C TEST NIN AGAINST NVAL
  150 IF(NVAL(I).NE.NIN(I)) TYPE 2030,I,W2(NIN(I)+1),
  $ W3(NIN(I)+1),W2(NVAL(I)+1),W3(NVAL(I)+1)
2030 FORMAT(1H0,'*****WARNING : VALVE ',I2,' IS ',2A3,
  $ ' .....THIS VALVE MUST BE ',2A3)
  190 NTEST(I) = 0
  110 CONTINUE
  RETURN
  END
```

```
SUBROUTINE VBVK19(J)
COMMON/BD7/ FLM1,FLM2,FLM3,FLM4,FLM5,FLM6,FLM7,IJPM(7)
C JUMP TO REQUIRED LINE
GOTO(10,20,30),IJPM(J)
C DIAFRAGM
10 TYPE 9010
RETURN
C PENETRATION LENGTH
20 TYPE 9020
RETURN
C NUMBER OF TURNS
30 TYPE 9030
9010 FORMAT(1H$, 'DIAFRAGM (MM)')
9020 FORMAT(1H$, 'PENETRATION LENGTH (MM)')
9030 FORMAT(1H$, 'NUMBER OF TURNS')
RETURN
END
```

```
SUBROUTINE VBVK20
COMMON/GEN/NIN(64),NVAL(128),DI(24),DO(4),ACTI,
$ IERROR,IFLAG1,IFLAG2,IFLAG3,IFLAG4
COMMON/BB3/NTIM(50),NDIO(50),TI(10),NSIZE,NFL1
COMMON/DD1/ ELAP,DMAS(7),STORE(290)
C ROUTINE WHICH CONTROLS THE VALVE OPERATION DURING THE TEST
C CALL FOR TIME
T1 = SECNDS(0.0)
ACTO = -ELAP
C SET IFLAG4 VBVK30 CALLED ONLY BY VBVK20
IFLAG4 = 1
DO 10 I=1,NSIZE
ACTI = TI(NTIM(I)+1)
C LOOP OVER THE REQUIRED TIME SPACING
```

C READ AND WRITE INTERFACES

```
15 CALL VBVK30(1)
    CALL VBVK30(2)
    CALL VBVK30(3)
    CALL VBVK30(4)
    DELTA = SECNDS(T1)
```

C STORE A/D AND D/A INFORMATION IN STORE(EACH ELAP SECONDS)

```
IF(DELTA.LT.(ACTO+ELAP)) GOTO 13
```

C ELAP SECONDS LATER...

C EXIT IN CASE OF MORE STORAGE THAN POSSIBLE

```
IF(IFLAG2.GE.10.AND.IERROR.EQ.0) IERROR = 20001
```

```
IF(IERROR.NE.0) RETURN
```

```
STORE(29*IFLAG2+1) = DI(1) * 1.0E-05
```

```
STORE(29*IFLAG2+2) = DI(2) * 1.0E-05
```

```
STORE(29*IFLAG2+3) = DI(3)
```

```
STORE(29*IFLAG2+4) = DO(1)
```

```
STORE(29*IFLAG2+5) = DI(4) * 1.0E-05
```

```
STORE(29*IFLAG2+6) = DI(5)
```

```
ACTO = ELAP * AINT(DELTA/ELAP)
```

```
STORE(29*IFLAG2+7) = DELTA
```

```
STORE(29*IFLAG2+8) = DI(6)
```

```
STORE(29*IFLAG2+9) = DI(20) * 1.0E-05
```

```
STORE(29*IFLAG2+10) = DI(21)
```

```
STORE(29*IFLAG2+11) = DI(7) * 1.0E-05
```

```
STORE(29*IFLAG2+12) = DI(8) * 1.0E-05
```

```
STORE(29*IFLAG2+13) = DI(9)
```

```
STORE(29*IFLAG2+14) = DO(2)
```

```
STORE(29*IFLAG2+15) = DI(10) * 1.0E-05
```

```
STORE(29*IFLAG2+16) = DI(11) * 1.0E-05
```

```
STORE(29*IFLAG2+17) = DMAS(3)
```

```
STORE(29*IFLAG2+18) = DI(12) * 1.0E-05
```

```
STORE(29*IFLAG2+19) = DMAS(4)
```

```
STORE(29*IFLAG2+20) = DI(13) * 1.0E-05
```

```
STORE(29*IFLAG2+21) = DI(14) * 1.0E-05
```

```
STORE(29*IFLAG2+22) = DMAS(5)
```

```
STORE(29*IFLAG2+23) = DI(15) * 1.0E-05
```

```
STORE(29*IFLAG2+24) = DI(16) * 1.0E-05
```

```
STORE(29*IFLAG2+25) = DI(17)
STORE(29*IFLAG2+26) = DO(3)
STORE(29*IFLAG2+27) = DI(18) * 1.0E-05
STORE(29*IFLAG2+28) = DI(19) * 1.0E-05
STORE(29*IFLAG2+29) = DMAS(7)
IFLAG2 = IFLAG2 + 1
13 CONTINUE
IF(IERROR.NE.0) RETURN
IF(DELTA.LT.ACTI) GOTO 15
C TAKE ACTION; ACTIVATE OR RELEASE BITS
NV1 = ISIGN(1,NDIO(I))
IF(NV1.LE.0) NV1 = 0
NVAL(IABS(NDIO(I))) = NV1
NVAL(IABS(NDIO(I))-72) = NV1
C CONTROL LIGHTS : VITIATOR AND SFCC
IF(IABS(NDIO(I)).EQ.118) GOTO 20
IF(IABS(NDIO(I)).EQ.119) GOTO 25
C BITS 109 .. 111 INDICATE PURGE FASE
IF(IABS(NDIO(I)).GE.109.AND.IABS(NDIO(I)).LE.111)
$ GOTO 35
GOTO 40
C IGNITION VITIATOR AND HOT AIR SUPPLY
20 NVAL(70) = 0
NVAL(71) = NV1
NVAL(72) = 1
GOTO 40
C IGNITION SFCC AND SFCC RUN
25 NVAL(70) = 0
NVAL(73) = NV1
NVAL(74) = 1
GOTO 40
C PURGE FASE, ALSO END OF TEST
35 CONTINUE
NVAL(72) = 0
NVAL(74) = 0
NVAL(75) = NV1
40 CONTINUE
```

C READ AND WRITE INTERFACES SECOND TIME

CALL VBVK30(1)

CALL VBVK30(2)

CALL VBVK30(3)

CALL VBVK30(4)

10 CONTINUE

RETURN

END

SUBROUTINE VBVK30(L)

COMMON/GEN/NIN(64),NVAL(128),DI(24),DO(4),ACTI,

\$ IERROR,IFLAG1,IFLAG2,IFLAG3,IFLAG4

COMMON/IJK/AIJK(21),BIJK(21),CIJK(3),DIJK(3)

C ROUTINE FOR BIT MANIPULATION AND A/D,D/A HANDLING

C L=1 READ OCTAL REPRESENTATION IN NWORD AND TRANSFORM TO NBIT

C L=2 READ NBIT AND TRANSFORM TO OCTAL REPRESENTATION IN NWORD

C L=3 READ A/D CONVERTER

C L=4 WRITE D/A CONVERTER

INTEGER VZERO

DIMENSION NWORD(4),NCHAN(16)

DIMENSION NJ(8),NC(8),NI(8)

C NJ,NC AND NI PERFORM THE TRANSFORMATION FROM THE

C LATEST WIRING TO THE DESIRED WIRING SEE L=1

DATA NJ/1,1,2,2,3,3,4,4/

\$ NC/0,8,0,8,0,8,0,8/

\$ NI/56,32,40,48,0,8,16,24/

C NCHAN CONTAINS OCTAL REPRESENTATION OF BIT SETTING

DATA NCHAN/"1,"2,"4,"10,"20,"40,"100,"200,"400,

\$ "1000,"2000,"4000,"10000,"20000,"40000,

\$ "100000/

VZERO = "4000

C CHAIN TO WANTED OPTION

GOTO(1,2,3,4),L

C -----

C L=1 TRANSFORM NWORD TO NBIT

C INITIALISE NBIT

1 CONTINUE

C INPUT CALL DJIN

DO 5 I=1,4

CALL DJIN(NWORD(I),I-1)

5 CONTINUE

C CODING IS AS FOLLOWS

C NIN(1..32) = PORT C + D

C NIN(33..40) = PORT A BIT 9..16

C NIN(41..56) = PORT B

C NIN(57..64) = INTERRUPT BITS PORT A BIT 1..8

C SCAN NWORD FOR BIT SETTING

DO 25 J=1,8

DO 20 I=1,8

IF((NWORD(NJ(J)).AND.NCHAN(NC(J)+I)) .EQ.

\$ NCHAN(NC(J)+I)) GOTO 30

NIN(I+NI(J)) = 0

GOTO 20

30 NIN(I+NI(J)) = 1

20 CONTINUE

25 CONTINUE

RETURN

C -----

2 CONTINUE

C L=2 TRANSFORM NVAL(65..128) TO NWORD

C INITIALISE NWORD

DO 100 I=1,4

NWORD(I) = "0

100 CONTINUE

C SCAN NBIT FOR BIT SETTING

DO 250 J=1,4

DO 200 I=1,16

IF(NVAL(I+48+16*J).EQ.1) NWORD(J) = NWORD(J).OR.NCHAN(I)

200 CONTINUE

250 CONTINUE

C OUTPUT CALL DJOUT

DO 260 I=1,4

```
      CALL DJOUT(NWORD(I),I+3)
260 CONTINUE
      RETURN
C -----
      3 CONTINUE
C L=3 READ A/D CONVERTER
      DO 300 I=1,21
C CONVERT TO VOLT
      DUM = FLOAT((IADC(I-1)-VZERO)/400.0) / 1.025
C CONVERT TO REAL PRESSURES AND TEMPERATURES
      DI(I) = DUM * AIJK(I) + BIJK(I)
300 CONTINUE
      RETURN
C -----
      4 CONTINUE
C L=4 WRITE D/A CONVERTER
      DO 400 I=1,3
C CONVERT DA OUTPUT TO VOLT
      J = IFIX((5.0 - DO(I)*CIJK(I) + DIJK(I)) * 409.5)
      CALL DAC(I-1,J)
400 CONTINUE
      RETURN
C -----
      END
```

```
      SUBROUTINE VBVK40
C ROUTINE FOR PRINT OUT OF PRESSURES,TEMPERATURES AND MASS FLOWS
      COMMON/GEN/NIN(64),NVAL(128),DI(24),DO(4),ACTI;
      $          IERROR,IFLAG1,IFLAG2,IFLAG3,IFLAG4
      COMMON/DD1/ ELAP,DMAS(7),STORE(290)
      IF(IFLAG2.LE.0) RETURN
C TYPE HEADING
      TYPE 1000
C TYPE STORE
      IFLAG2 = 29*IFLAG2
```

```
TYPE 1010, (STORE(I), I=1, IFLAG2)
1000 FORMAT(1H0, 'LINE', 7X, 'P(PRIM;BAR)', 1X, 'P(SEC;BAR)', 5X,
$         'T(K)', 6X, 'M(KG/S)', 5X, 'LINE', 7X,
$         'P(PRIM;BAR)', 1X, 'P(SEC;BAR)', 5X, 'T(K)', 6X,
$         'M(KG/S)', 8X, 'TIME(S)')
1010 FORMAT(1X, '*****', /,
$         1X, 'AIR', 7X, 4F12.4, 2X, 'VITIATOR', 14X, 2F12.4, 12X,
$         F12.4,
$         /, 1X, 'PILOT', 29X, F12.4, 14X, 'SFCC', 18X, 2F12.4,
$         /, 1X, 'O2(1)', 5X, 4F12.4, 2X, 'O2(2)', 5X, 2F12.4, 12X, F12.4,
$         /, 1X, 'O2(3)', 17X, F12.4, 12X, F12.4, 2X, 'H2', 8X, 2F12.4,
$         12X, F12.4,
$         /, 1X, 'CH4(1)', 4X, 4F12.4, 2X, 'CH4(2)', 4X, 2F12.4,
$         12X, F12.4)
```

C RESET IFLAG2

IFLAG2 = 0

RETURN

END

SUBROUTINE VBVK50(X,Y,N)

C SORTING X WITH BUBBLE-SORT METHOD

DIMENSION X(N),Y(N)

IF(N.LE.1) RETURN

DO 200 I=1,(N-1)

C THE LAST I-1 NUMBERS ARE ALREADY SORTED

DO 100 J=1,(N-I)

IF(X(J).LE.X(J+1)) GOTO 100

C CHANGE X(J) AND X(J+1)

XX = X(J+1)

X(J+1) = X(J)

X(J) = XX

XX = Y(J+1)

Y(J+1) = Y(J)

Y(J) = XX

100 CONTINUE

```
200 CONTINUE
RETURN
END

SUBROUTINE VBVK99(I)
COMMON/GEN/NIN(64),NVAL(128),DI(24),DO(4),ACTI,
$ IERROR,IFLAG1,IFLAG2,IFLAG3,IFLAG4
C ERROR ROUTINE
C RESET CONTROL O ACTION
CALL RCTRLO
I = IABS(IERROR)
IF(IERROR.LE.0) RETURN
C ACTIVATE SECURITY SYSTEM FOR IERROR.GT.0
NVAL(125) = 1
J = IERROR / 1000
TYPE 1000,J
1000 FORMAT(1H0,'FATAL ERROR IN VBVK',I2)
J = IERROR - J * 1000
TYPE 1010,J
1010 FORMAT(1H0,'ERROR CODE.....',I3)
C PRINT IMPORTANT QUANTITIES
10 TYPE 1020,IERROR,IFLAG1,IFLAG2,IFLAG3,IFLAG4,
$ (NIN(J),J=1,64),(NVAL(J),J=1,64),
$ (NVAL(J),J=65,128)
1020 FORMAT(1H0,'FLAGS.....',5I9,/,
$ 1H,'INPUT BITS..',64I1,/,
$ 1H,'CONTROL BITS..',64I1,/,
$ 1H,'OUTPUT BITS..',64I1)
RETURN
END
BLOCK DATA
C DATA FOR CALCULATION OF MASS FLOW AND PENETRATION LENGTH
COMMON/BD1/TO(4),CO(4),C1(4),C2(4),C3(4),C4(4)
C DATA FOR CALCULATION OF THE PRESSURE DECREASINGS DURING
C TEST
```

COMMON/BD2/ GM(7),V(18),VL(7),NPB(18),NBB(7)
COMMON/BD3/ AMARK1(7),AMARK2(7),NPP(7),NPS(7),NTT(7)
COMMON/BD4/ TNUL(7),PL(3,7),Z1(7),Z2(7),Z3(7),Y1(7),Y2(7)
C COMMON/BD5/NS(5),NTH(12),NTSCI(14),NTSH(22),NTSHI(32),
C \$ NTSECI(16),NT(5),NVH(12),NVSCI(14),NVSH(22),
C \$ NVSHI(32),NVSECI(16),TVL1,TVL2,TVL3,TVL4
C \$,TVL5
COMMON/BD5/NS(5),NTH(12),NTSCI(12),NTSH(22),NTSHI(32),
\$ NTSECI(16),NT(5),NVH(12),NVSCI(12),NVSH(22),
\$ NVSHI(32),NVSECI(16),TVL1,TVL2,TVL3,TVL4
\$,TVL5
COMMON/BD6/NST(6),NV(100)
COMMON/BD7/ FLM1,FLM2,FLM3,FLM4,FLM5,FLM6,FLM7,IJPM(7)
C DATA FOR CALIBRATION OF THE PRESSURES AND TEMPERATURES
COMMON/IJK/AIJK(21),BIJK(21),CIJK(3),DIJK(3)
DATA T0/240.0,270.0,300.0,330.0/
DATA C0/-4.71894,-5.3445,-5.91378,-6.51733/
DATA C1/19.5882E-03,19.8122E-03,19.6724E-03,19.7893E-03/
DATA C2/-3.24382E-07,-9.68416E-07,-8.21302E-07,-1.33002E-06/
DATA C3/28.2324E-10,34.8811E-10,34.6886E-10,39.954E-10/
DATA C4/-7.23437E-13,-9.68372E-13,-1.00945E-12,-1.18238E-12/
C GM MOLECULAIR MASS OF LINES 1..7
C V VOLUMES OF BUFFERS,SEE ALSO ARRAY NPB
C VL VOLUMES OF PIPES
C NPB ARRAY WHICH GIVES WITH AID OF ARRAY NBB THE VALVE
C NUMBERS OF THE BUFFERS PER LINE
C NBB NUMBER OF BUFFERS PER LINE
DATA GM/28.96,3*32.0,2.02,2*16.04/
DATA V/3.024,7*0.05,4*0.001,2*0.05,0.01,2*0.05,0.01/
DATA VL/7*1.0E-05/
DATA NPB/1,3,4,5,8,9,8,9,13,14,15,16,19,20,21,19,20,21/
DATA NBB/1,3,2,2,4,3,3/
C AMARK1 AND AMARK2 ARE THE ID'S FOR THE SEVEN LINES
C NPP : A/D CHANNEL NUMBER FOR THE PRIMARY PRESSURE PER LINE
C NPS : A/D CHANNEL NUMBER FOR THE SECONDARY PRESSURE
C NTT : A/D CHANNEL NUMBER FOR THE TEMPERATURE
DATA AMARK1/3HAIR,3HO2(,3HO2(,3HO2(,3HH2 ,

```
$          3HCH4,3HCH4/
DATA AMARK2/3H  ,3H1) ,3H2) ,3H3) ,3H  ,
$          3H(1),3H(2)/
DATA NPP/1,7,10,10,13,15,18/,NPS/2,8,11,12,14,16,19/,
$          NTT/3,9,9,9,9,17,17/
C CONSTANTS FOR SCMC'S
C TNUL      : REF. TEMPERATURE
C          = 0.0 : NO CORRECTION
C PL(2,J)   : MINIMUM PENETRATION LENGTH IN MM
C          OR MINIMUM DIAFRAGM IN MM
C          OR MINIMUM NUMBER OF TURNS
C PL(3,J)   : MAXIMUM PENETRATION LENGTH IN MM
C          OR MAXIMUM DIAFRAGM IN MM
C          OR MAXIMUM NUMBER OF TURNS
C Z1,Z2,Z3: POLYNOMIAL CONSTANTS FOR AT VS PEN.LENGTH
C Y1,Y2     : MASS/AT VS P CURVE
DATA TNUL/0.0,300.0,5*0.0/
DATA PL(2,1),PL(2,2),PL(2,3),PL(2,4),PL(2,5),PL(2,6),PL(2,7)/
$      4.5 , 50.0 , 0.05 , 0.05 , 0.00 , 4.5 , 0.00/
DATA PL(3,1),PL(3,2),PL(3,3),PL(3,4),PL(3,5),PL(3,6),PL(3,7)/
$      10.0 ; 69.0 , 1.00 , 1.00 , 1.00 , 10.0 , 1.00/
DATA Z1/0.7854,-.00591,-76.03,-76.03,-.6416,0.7854,-76.03/
DATA Z2/0.0,-.17707,2*205.78,1.7052,0.0,205.78/
DATA Z3/0.0,46.30,2*-10.08,0.2246,0.0,-10.08/
DATA Y1/2.7744E-09,2.451E-09,3*1E-9,2.7744E-09,1E-9/
DATA Y2/0.443258E-03,0.0349E-03,3*0.0,0.443258E-03,0.0/
C CONSTANTS FOR VBVK12
C NS GIVES INFORMATION ABOUT THE LENGTH OF THE ARRAYS
C NTH,NTSCI,NTSH,NTSHI,NTSECI AND NVH,NVSCI,NVSH,NVSHI,
C NVSECI
C ARRAYS BEGINNING WITH NT GIVE INFO ABOUT THE TIMING
C ARRAYS BEGINNING WITH NV GIVE INFO ABOUT THE VALVES
C TIMING ARRAYS GIVE INFO OVER TIMEPOINTS E.G. T(0),T(1),
C T(2), ETC...
C VALVES ARRAYS GIVE INFO ABOUT BIT NUMBERS, - IS 0, + IS
C 1.
C EACH TEST HAS ITS OWN ARRAYS
```

C DATA NS/12,14,22,32,16/
 DATA NS/12,12,22,32,16/
 DATA NT/6, 6, 8,10, 8 /
 DATA NTH /1,1,2,2,2,3,3,4,4,4,5,5/
 C DATA NTSCI /0,1,1,1,1,2,3,3,3,4,4,4,5,5/
 DATA NTSCI /1,1,1,1,2,3,3,3,4,4,5,5/
 DATA NTSH /1,1,1,1,2,2,2,2,3,3,3,4,5,5,5,6,6,6,7,7,7,7/
 DATA NTSHI /1,1,1,1,2,2,2,2,3,3,3,4,4,4,4,5,6,6,6,7,8,8,8,8,
 \$ 8,8,9,9,9,9,9,9/
 DATA NTSECI/1,1,2,2,2,2,3,4,4,4,5,5,6,6,7,7/
 DATA NVH /98,100,103,104,119,-103,-119,-98,111,-100,-111,-104/
 C DATA NVSCI /97,102,103,104,119,98,-102,-103,-119,-98,-97,
 C \$ 111,-111,-104/
 DATA NVSCI /102,103,104,119,97,-102,-103,-119,-97,
 \$ 111,-111,-104/
 DATA NVSH /101,107,108,118,97,100,105,106,-101,-107,-118,
 \$ 98,-98,109,110,-97,-100,-105,-109,-110,106,-108/
 DATA NVSHI /101,107,108,118,97,100,105,106,-101,-107,-118,
 \$ 102,103,104,119,98,-102,-103,-119,-98,-97,-100,
 \$ -105,109,110,111,-111,-110,-109,-106,-108,-104/
 DATA NVSECI/97,100,102,103,104,119,98,-102,-103,-119,-98,
 \$ 111,-97,-100,-111,-104/

C TVL1...TVL5 ARE DELAY TIMES FOR RESP. TO T1,

C T2 TO T3,PURGE TIME, AND T6 TO T7

DATA TVL1/1.0/,TVL2/0.5/,TVL3/15.0/,TVL4/2.5/,TVL5/30.0/

C NST GIVES THE ADDRESS IN NV FOR A PARTICULAR GAS LINE

DATA NST/ 1, 6,19,31,44,58/

C NV GIVES PER LINE THE REQUIRED SETTINGS OF THE VALVES

C SUPERSEDED BY 900 MEANS AN OPTIONAL SETTING

C SUPERSEDED BY 999 MEANS A REQUIRED SETTING

C EACH LINE IS SUPERSEDED BY THE TOTAL NUMBER OF DATA

C FOR THE LINE

DATA NV/4,999,-1,999,1,

\$ 12,999,-6,999,2,900,3,4,5,999,-2,999,6,

\$ 11,999,-10,999,7,900,8,9,999,-7,999,10,

\$ 12,999,-10,999,7,900,8,9,999,-7,999,10,11,

\$ 13,999,-17,999,12,900,13,14,15,16,999,-12,

\$ 999,17,
\$ 12,999,-22,999,18,900,19,20,21,999,-18,999,
\$ 22,
\$ 30*999/

C FLM1...FLM4 ARE CONSTANTS FOR CALCULATING THE MASS-FLOWS

C FLM1 = FACTOR TO OBTAIN H2 FLOW = 0.125

C FLM2 = MAXIMUM H2 FLOW

C FLM3 = PERCENTAGE FOR NON SCMC O2 LINES

C FLM4 = CH4 LINES

DATA FLM1/0.125/,FLM2/0.012/,FLM3/0.05/,FLM4/0.167/

C FLM5 IS THE MAXIMUM ALLOWABLE SECONDARY PRESSURE

DATA FLM5/80.0E+05/

C FLM6 = CONSTANT MASSFLOW O2(2) FOR SH,SEH,SHI AND SEHI TEST

C FLM7 * APREL MUST BE LESS THAN SECONDARY PRESSURE

DATA FLM6/0.015/,FLM7/2.0/

C IJPM GIVES PRINT OUT FOR RESP. DIAFRAGM,PENETRATION AND

C NUMBER OF TURNS FOR RESP.1,2 AND 3

DATA IJPM/1,2,3,3,3,1,3/

C CONSTANTS FOR PRESSURES AND TEMPERATURES

DATA AIJK/4.0E+06,2.0E+06,80.0,2.0E+06,80.0,80.0,4.0E+06,

\$ 2.0E+06,80.0,4.0E+06,2.0E+06,2.0E+06,4.0E+06,

\$ 2.0E+06,4.0E+06,2.0E+06,80.0,4.0E+06,2.0E+06,

\$ 2.0E+06,80.0/

DATA BIJK/21*0.0/

C CONSTANTS FOR DA OUTPUT : VOLT = CIJK * MASS FLOW + DIJK

C I.E. 0 VOLT = 0 KG/S AND 5 VOLT = 2 KG/S

DATA CIJK/2.5,2.5,2.5/

DATA DIJK/0.0,0.0,0.0/

END


```
      SUBROUTINE VBLK01(PROG)
C INITIALIZEER UNIT NUMMERS
C ZET DE PROGRAMMA NAAM PROG(2) IN PR(2)
C HET UNITNUMMER VAN DE PRINTFILE IS NEGATIEF, ZOLANG DEZE
C NIET IS GEOPEND
C ZIE VOOR DE BETEKENIS VAN DE VARIABELE KPR SUBROUTINE VBLK02
      COMMON /SYS/ IIN, IOUT, IPR, KPR, IF1, IF2, IF3, IF4, PR(2)
      DIMENSION PROG(2)
      IIN = 5
      IOUT = 5
      IPR = -9
      KPR = IPR
      IF1 = 2
      IF2 = 3
      IF3 = 4
      IF4 = 8
      PR(1)=PROG(1)
      PR(2)=PROG(2)
      RETURN
      END
```

```
      SUBROUTINE VBLK02
C VRAAGT WAAR DE PRINT-UITVOER VAN HET PROGRAMMA NAARTOE MOET
C INDIEN NODIG WORDT EEN FILE GEOPEND MET ALS NAAM:
C (PROGRAMMA NAAM) .TXT
C NADAT EEN PRINT-FILE IS GEOPEND, WORDT SUBROUTINE VBLK03 AANGEROEPEN
C VOOR HET PRINTEN VAN EEN HEADER
C DE VARIABELE KPR BEVAT DE OORSPRONKELIJKE (NEGATIEVE) WAARDE
C VAN IPR (ZIE SUBR. VBLK01), OF DE ABSOLUTE WAARDE DAARVAN INDIEN
C DE FILE REEDS GEOPEND IS
C BIJ HET MEERDERE MALEN OPROEPEN VAN DE SUBR. GELDT DAT DE KEUZE
C TUSSEN (LINE-PRINTER +FILE) EN (FILE ALLEEN) SLECHTS EENMAAL
C GEMAAKT KAN WORDEN; KEUZE TUSSEN DE TERMINAL EN EEN VAN
C BOVENGENOEMDE MOGELIJKHEDEN BLIJFT MOGELIJK
      COMMON /SYS/ IIN, IOUT, IPR, KPR, IF1, IF2, IF3, IF4, PR(2)
```

```
DIMENSION FLN(7),PROG(2)
DATA FEXT/4H.TXT/, ENIL/4HLINE/, TERM/4HTERM/
IF(KPR.GT.0) GOTO 5
WRITE(IOUT,100)PR,FEXT
100 FORMAT(29HOCOMplete PRINT-OUT ON FILE ',2A3,A4,
* 27H' ? GIVE: 'FILE' (DEFAULT)/4X,4HOR: ,
* 48HCOMplete PRINT-OUT ON LINEPRINTER ? GIVE: 'LINE' /
* 21X,4HOR: ,31HONLY ON TERMINAL ? GIVE: 'TERM' /3H$ *)
GOTO 6
5 WRITE(IOUT,110)
110 FORMAT(42HOPRINT-OUT ONLY ON TERMINAL ? GIVE: 'TERM' /
* 3H$ *)
6 READ(IIN,200) ANTW
200 FORMAT(A4)
C MODE=1: UITVOER OP FILE; MODE=2: UITVOER OP FILE EN OP DE LINE-
C PRINTER (LPO); MODE=3: UITVOER OP DE TERMINAL
C ALS AL EEN FILE IS GEOPEND, DAN ZIJN 1 EN 2 IDENTIEK
MODE = 1
IF(ANTW.EQ.ENIL) MODE = 2
IF(ANTW.EQ.TERM) MODE = 3
GOTO (10,10,90), MODE
10 IF(KPR.GT.0) GOTO 80
IPR = -KPR
KPR = IPR
ENCODE(11,300,FLN) PR,FEXT
300 FORMAT(2A3,A4)
C VERWIJDER BLANKS UIT ARRAY EN SLUIT LINKS AAN EN SLUIT AF MET
C ASCII NUL KARAKTERS
CALL VBLK04(FLN)
GOTO (20,30), MODE
20 OPEN (UNIT=IPR,NAME=FLN,CARRIAGECONTROL='LIST')
GOTO 70
30 OPEN (UNIT=IPR,NAME=FLN,DISP='PRINT',
* CARRIAGECONTROL='LIST')
70 PROG(1)=PR(1)
PROG(2)=PR(2)
CALL VBLK03(PROG)
```

```
GOTO 99
80 IPR = KPR
   GOTO 99
90 IPR = IOUT
99 RETURN
   END
```

```
   SUBROUTINE VBLK03(PROG)
C BIJ DE EERSTE AANROEP:
C PRINT EEN HEADING WAARIN: 'PROGRAMMA' EN PR(2), FMT(2A3),
C                               'SEKTIE RAKETVOORTSTUWING'
C                               'DATUM' EN DATUM(3)
C BIJ EEN VOLGENDE AANROEP:
C PRINT EEN HEADING WAARIN: 'SUBPROGRAMMA' EN PROG(2) (2A3)
C GEBRUIKT SYSTEEM SUBROUTINE DATE (PDP11/45)
   COMMON /SYS/ IIN,IOUT,IPR,KPR,IF1,IF2,IF3,IF4,PR(2)
   DIMENSION TITLE(20),DATUM(3),PROG(2),PRG(3),SPRG(3),EMA(2)
   DATA S/4H****/, B/4H   /, LINE/1H0/, IEEN/0/,
* PRG/4HPRG,4HRAM ,4H   /, SPRG/4HSUBP,4HROGR,4HAM /
   IF(IEEN.NE.0) GOTO 10
   EMA(1)=PR(1)
   EMA(2)=PR(2)
   GOTO 15
10 WRITE(IPR,7)
   7 FORMAT(1H1)
   EMA(1)=PROG(1)
   EMA(2)=PROG(2)
15 WRITE(IPR,1) LINE, (S,I=1,13)
   1 FORMAT(A1,20A4)
   WRITE(IPR,1) B,S,(B,I=1,11),S
   WRITE(IPR,2) B,S,B,B,PRG,B,B,EMA,B,B,S
   2 FORMAT(A1,8A4,2X,2A3,3A4)
   IF(IEEN.NE.0) GOTO 20
   WRITE(IPR,1)B,S,(B,I=1,11),S
   WRITE(IPR,3)B,S,B,B,B,B,S
```

```
3  FORMAT(A1,3A4,28HSEKTIE      RAKETVOORTSTUWING,3A4)
   WRITE(IPR,1)B,S,(B,I=1,11),S
C  SYSTEEM SUBROUTINE DATE      (PDP11/45)
   CALL DATE(DATUM)
   WRITE(IPR,4)B,S,B,B,B,B,DATUM,B,B,S
4  FORMAT(A1,3A4,8HDATUM      ,2A4,3X,2A4,A1,3A4)
20 WRITE(IPR,1) B,S,(B,I=1,11),S
   WRITE(IPR,1) B,(S,I=1,13)
   WRITE(IPR,1) LINE
   WRITE(IOUT,5) PRG, EMA
5  FORMAT(41HOGIVE COMMENTS (IF DESIRED) FOR THIS RUN /
* 9H OF THE ,3A4,1X,2A3,24H (ONE LINE ONLY(20A4)) /)
   READ(IIN,6) TITLE
6  FORMAT(20A4)
   WRITE(IPR,1) LINE,TITLE
   WRITE(IPR,1) LINE
   IF(IEEN.NE.0) GOTO 99
   DO 25 I=1,3
25 PRG(I)=SPRG(I)
   IEEN=1
   IF(PR(1).NE.PROG(1).OR.PR(2).NE.PROG(2)) GOTO 10
99 RETURN
   END
```

SUBROUTINE VBLK04(BR)

```
C  VERWIJDEERT BLANKS UIT EEN REAL-ARRAY MET EEN DIMENSIE 7;
C  DE TEKST ZONDER BLANKS WORDT LINKS AANGESCHOVEN IN DE ARRAY
C  EN AFGESLOTEN MET EEN ASCII NUL KARAKTER
   DIMENSION AR(7),BR(7)
   BYTE IPAR(28),IB,NUL
   EQUIVALENCE (IPAR(1),AR(1))
C 32 IS DE ASCII WAARDE VAN EEN BLANK
   DATA IB/32/, NUL/0/
   DO 2 I=1,7
2  AR(I)=BR(I)
```

```
K=0
DO 1 I=1,28
  IF(IPAR(I).EQ.IB) GOTO 1
  K = K + 1
  IPAR(K)=IPAR(I)
1 CONTINUE
  K = K + 1
  IF(K.EQ.28) K=27
  DO 4 I=K,28
4 IPAR(I) = NUL
  DO 3 I=1,7
3 BR(I)=AR(I)
  RETURN
  END
```

PROGRAM VBLOK

C VBLOK PRINTS ALL DATA STORED IN BLOCK DATA

```
COMMON/SYS/IIN, IOUT, IPR, KPR, IF1, IF2, IF3, IF4, PR(2)
COMMON/BD1/TO(4), CO(4), C1(4), C2(4), C3(4), C4(4)
COMMON/BD2/ GM(7), V(18), VL(7), NPB(18), NBB(7)
COMMON/BD3/ AMARK1(7), AMARK2(7), NPP(7), NPS(7), NTT(7)
COMMON/BD4/ TNUL(7), PL(3,7), Z1(7), Z2(7), Z3(7), Y1(7), Y2(7)
C COMMON/BD5/NS(5), NTH(12), NTSCI(14), NTSH(22), NTSHI(32),
C $ NTSECI(16), NT(5), NVH(12), NVSCI(14), NVSH(22),
C $ NVSHI(32), NVSECI(16), TVL1, TVL2, TVL3, TVL4
C $ , TVL5
COMMON/BD5/NS(5), NTH(12), NTSCI(12), NTSH(22), NTSHI(32),
$ NTSECI(16), NT(5), NVH(12), NVSCI(12), NVSH(22),
$ NVSHI(32), NVSECI(16), TVL1, TVL2, TVL3, TVL4
$ , TVL5
COMMON/BD6/NST(6), NV(100)
COMMON/BD7/ FLM1, FLM2, FLM3, FLM4, FLM5, FLM6, FLM7, IJPM(7)
C DATA FOR CALIBRATION OF THE PRESSURES AND TEMPERATURES
COMMON/IJK/AIJK(21), BIJK(21), CIJK(3), DIJK(3)
DIMENSION VTOT(7)
```

```
DIMENSION PROG(2)
DATA PROG/3HVBL,3HOK /
C CALL FOR POSSIBLE PRINT-OUTPUT
CALL VBLK01(PROG)
C GIVE PROGRAM HEADING ON TERMINAL
WRITE(IOUT,998)
CALL VBLK02
C GIVE A PRINT OUT OF BD1
WRITE(IPR,998)
998 FORMAT(1H0,
$'PROGRAM VBLOK, A PROGRAM TO PRINT OUT THE DATA STORED IN ',/,
$' THE BLOCK DATA OF THE SFCC PROGRAM',/,
$'OVERSION OF 22-12-83')
WRITE(IPR,999)
WRITE(IPR,1000)
999 FORMAT(1H1,72(1H*))
1000 FORMAT(
$'COMMON BLOCK BD1 IS DEFINED AS :',/,
$'COMMON/BD1/T0(4),C0(4),C1(4),C2(4),C3(4),C4(4).',/,
$'OTHESE CONSTANTS ARE USED TO CALCULATE THE RELATIVE MASSFLOWS
',/,
$' WHEN USING THE VITIATOR.',/,
$'OTHE METHOD OF CALCULATION IS GIVEN BY R.G.VAN BRUGGEN IN',/,
$' SFCC PUBLICATION 1.',/,
$'OTHE RELATIVE METHANE-MASSFLOW GIVEN IN TABLE 4 OF THIS PU-',/,
$' BLICATION IS APPROXIMATED BY A FOURTH ORDER POLYNOMIAL
WITH',/,
$' CONSTANTS C0,C1,C2,C3 AND C4, IN THE TEMPERATURE T(E).',/,
$'OFOR EACH TEMPERATURE T(0) A POLYNOMIAL IS GIVEN.')
```

```
WRITE(IPR,1010)
1010 FORMAT(1H0,'TEMP POLYNOMIAL CONSTANTS',/,
$          T2,'T0',T9,'C0',T24,'C1',T39,'C2',T54,
$          'C3',T69,'C4')
WRITE(IPR,1020)((T0(I),C0(I),C1(I),C2(I),C3(I),C4(I)),
$          I=1,4)
1020 FORMAT(1X,F4.0,5E15.5)
WRITE(IPR,1030)
```

```
1030 FORMAT(1H0,'FOR THIS TABLE :',/,/,
      $      1X,'TEMPERATURE IN KELVIN.')
```

C COMMON BLOCK BD2

```
      WRITE(IPR,999)
      WRITE(IPR,1100)
```

1100 FORMAT(
 \$'COMMON BLOCK BD2 IS DEFINED AS :',/,/,
 \$'COMMON/BD2/GM(7),V(18),VL(7),NPB(18),NBB(7).',/,/,
 \$'THIS COMMON BLOCK IS USED TO CALCULATE THE ACTUAL BUFFER-',/,/,
 \$' VOLUMES.',/,/,
 \$'OGM CONTAINS THE MOLECULAIR MASSES FOR THE USED GASES.',/,/,
 \$' NBB CONTAINS THE NUMBER OF BUFFERVESSELS PER USED LINE.',/,/,
 \$' VL CONTAINS THE VOLUMES OF THE PIPES PER LINE.')

```
      WRITE(IPR,1110)((AMARK1(I),AMARK2(I),GM(I),NBB(I),VL(I)),I=1,7)
```

1110 FORMAT(
 \$'OGAS MOLECULAIR MASS NUMBER OF VOLUME OF
PIPES',/,/,
 \$' .G/MOL BUFFERVESSELS M**3',/,/,
 \$(1X,2A3,5X,F10.5,T34,I2,12X,F10.5))

```
      WRITE(IPR,1120)
```

1120 FORMAT(
 \$'OTHE TOTAL NUMBER OF BUFFERVESSELS ON THIS LIST IS 18.',/,/,
 \$' FOR EACH VESSEL INFORMATION ABOUT THE VOLUME AND THE
NUMBER',/,/,
 \$' OF THE INPUTBIT IS GIVEN IN RESP. V AND NPB.')

```
      WRITE(IPR,1130)((I,V(I),NPB(I)),I=1,18)
```

1130 FORMAT(
 \$'OBUFFERVESSEL VOLUME INPUTBIT',/,/,
 \$' M**3',/,/,
 \$(5X,I2,T18,F10.4,7X,I2))

```
      WRITE(IPR,1140)
```

1140 FORMAT(
 \$'OWITH THE AID OF THE NBB,V AND NPB ARRAY WE CAN CALCULATE',/,/,
 \$' THE ACTUAL BUFFERVOLUME.',/,/,
 \$' THE TOTAL BUFFERVOLUME CAN BE CALCULATED FROM NBB AND V :')

```
      K = 0
      DO 110 J=1,7
```

```
      K = K + NBB(J)
      VTOT(J) = 0.0
      DO 120 I=1,NBB(J)
      VTOT(J) = VTOT(J) + V(K+I-NBB(J))
120 CONTINUE
110 CONTINUE
      WRITE(IPR,1150)((AMARK1(I),AMARK2(I),VTOT(I)),I=1,7)
1150 FORMAT(
      $'OGAS          TOTAL BUFFER VOLUME',/,
      $'              M**3',/,
      $(1X,2A3,12X,F10.5))
C COMMON BLOCK BD3
      WRITE(IPR,999)
      WRITE(IPR,1200)
1200 FORMAT(
      $'OCOMMON BLOCK BD3 IS DEFINED AS :',/,
      $'OCOMMON/BD3/AMARK1(7),AMARK2(7),NPP(7),NPS(7),NTT(7).',/,
      $'OAMARK1 AND AMARK2 CONTAIN INFORMATION ABOUT THE TYPE OF
GAS,',/,
      $' NPP CONTAINS THE A/D CHANNEL NUMBER FOR THE PRIMARY PRES-',/,
      $' SURE PER LINE,',/,
      $' NPS CONTAINS THE A/D CHANNEL NUMBER FOR THE SECONDAY PRES-',/,
      $' SURE PER LINE,',/,
      $' NTT CONTAINS THE A/D CHANNEL NUMBER FOR THE TEMPERATURE
PER',/,
      $' LINE.')
```

```
      WRITE(IPR,1210)((I,AMARK1(I),AMARK2(I),NPP(I),NPS(I),NTT(I)),
      $              I=1,7)
1210 FORMAT(
      $'OLINE  GAS          NPP      NPS      NTT',/,
      $'      (AMARK1+AMARK2)',/,
      $(2X,I2,3X,2A3,T29,I2,8X,I2,8X,I2))
C COMMON BLOCK BD4
      WRITE(IPR,999)
      WRITE(IPR,1300)
1300 FORMAT(
      $'OCOMMON BLOCK BD4 IS DEFINED AS :',/,
```



```
$'OCOMMON/BD4/TNUL(7),PL(3,7),Z1(7),Z2(7),Z3(7),Y1(7),Y2(7).',/,
$'OTHIS BLOCK CONTAINS INFORMATION ABOUT THE MASSFLOW
DEVICES.',/,
$'OTNUL CONTAINS THE CALIBRATION TEMPERATURE (ONLY FOR SCMC',/,
$' DEVICES), THE CALCULATED MASSFLOW WILL BE CORRECTED WITH',/,
$' SQRT(TNUL/ACTUAL TEMPERATURE).',/,
$'OPL(2,J) CONTAINS THE MINIMUM ACCEPTABLE DIAFRAGM (IN MM)',/,
$' OR PENETRATION LENGTH (IN MM) OR NUMBER OF TURNS.',/,
$' PL(3,J) CONTAINS THE MAXIMUM ACCEPTABLE DIAFRAGM (IN MM)',/,
$' OR PENETRATION LENGTH (IN MM) OR NUMBER OF TURNS.')
```

WRITE(IPR,1310)((AMARK1(I),AMARK2(I),TNUL(I),PL(2,I),PL(3,I)),

\$ I=1,7)

```
1310 FORMAT('OGAS      TNUL      PL(2,J)    PL(3,J)')',/,
$      '      K',/,
$(1X,2A3,3X,F5.1,9X,F5.2,6X,F5.2))
WRITE(IPR,1320)
1320 FORMAT(
$'OZ1,Z2 AND Z3 ARE THE CONSTANTS FOR A SECOND ORDER
POLYNOMIAL',/,
$' WHICH GIVES THE RELATION BETWEEN AT AND (DIAFRAGM/PENE-',/,
$' TRATION LENGTH) OR BETWEEN MASSFLOW/PRESSURE AND (NUMBER
OF',/,
$' TURNS) : ',/,
$'OAT OR MASSFLOW/PRESSURE = Z3 + Z2 * L + Z1 * L * L')
```

WRITE(IPR,1330)(I,IJPM(I),AMARK1(I),AMARK2(I),Z3(I),Z2(I),Z1(I),

\$ I=1,7)

```
1330 FORMAT(
$'OLINE  TYPE    GAS      Z3      Z2      Z1',/,
$'      DEVICE',/,
$(2X,I2,5X,I2,5X,2A3,3X,3F10.5))
WRITE(IPR,1335)
1335 FORMAT(1H0,'FOR THIS TABLE :',//,
$' L IN MM, AT IN MM**2, MASSFLOW IN G/S, PRESSURE IN MPA.')
```

WRITE(IPR,1340)

```
1340 FORMAT(
$'ODEVICE TYPE IS GIVEN IN COMMON BLOCK BD7 AND HAS THE FOL-',/,
$' LOWING MEANING :',/,
```

```
$'OTYPE 1 = DEVICE WITH FIXED DIAFRAGM',/,
$' TYPE 2 = SCMC DEVICE',/,
$' TYPE 3 = DEVICE WITH ADJUSTABLE NUMBER OF TURNS',/,
$'OY1 AND Y2 ARE CONSTANTS IN THE RELATION BETWEEN
MASSFLOW/AT',/,
$' AND PRESSURE FOR DEVICES WITH DIAFRAGMS AND PENETRATION-',/,
$' LENGTHS AND CONSTANTS IN THE RELATION BETWEEN MASSFLOW AND',/,
$' PRESSURE FOR DEVICES WITH NUMBER OF TURNS :',/,
$'MASSFLOW/AT OR MASSFLOW = Y1 + Y2 * PRESSURE')
WRITE(IPR,1350)(I,IJPM(I),AMARK1(I),AMARK2(I),Y1(I),Y2(I),I=1,7)
1350 FORMAT(
$'OLINE TYPE GAS Y1 Y2',/,
$' DEVICE',/,
$(2X,I2,5X,I2,5X,2A3,3X,E15.5,5X,E15.5))
WRITE(IPR,1360)
1360 FORMAT(1H0,'FOR THIS TABLE :',//,
$ ' MASSFLOW IN KG/S, PRESSURE IN PASCAL, AT IN MM**2.')
```

C COMMON BLOCK BD5

```
WRITE(IPR,999)
WRITE(IPR,1400)
1400 FORMAT(
$'OCOMMON BLOCK BD5 IS DEFINED AS :',/,
$'OCOMMON/BD5/NS(5),NTH(12),NTSCI(12),NTSH(22),NTSHI(32)',/,
$' NTSECI(16),NT(5),NVH(12),NVSCI(12),NVSH(22)',/,
$' NVSHI(32),NVSECI(16),TVL1,TVL2,TVL3,TVL4,TVL5')
WRITE(IPR,1410)
1410 FORMAT(
$'ONS GIVES INFORMATION ABOUT THE LENGTH OF THE NTH/NVH',/,
$' NTSCI/NVSCI,NTSH/NVSH,NTSHI/NVSHI AND NTSECI/NVSECI ARRAYS',/,
$' RESPECTIVELY.')
```

```
WRITE(IPR,1420)((NS(I)),I=1,5)
1420 FORMAT(
$'OARRAY NUMBER OF ELEMENTS',//,
$' NTH,NVH ',I2,/,
$' NTSCI,NVSCI ',I2,/,
$' NTSH,NVSH ',I2,/,
$' NTSHI,NVSHI ',I2,/,
```

```
    $' NTSECI,NVSECI      ',I2)
      WRITE(IPR,1430)
1430 FORMAT(
    $'OALL THESE ARRAY ARE USED TO SET UP THE TIMING OF THE
VALVES, ',/,
    $' THE SECOND CHARACTER OF THE ARRAY NAME INDICATES WHETHER ',/,
    $' THE ARRAY IS USED AS TIMING (T) ARRAY OR AS VALVE (V)
ARRAY. ',/,
    $' CHARACTERS 3 TILL END ARE USED TO INDICATE THE TYPE OF THE ',/,
    $' EXPERIMENT. ')
      WRITE(IPR,1440)
```

```
1440 FORMAT(
    $'OTHE TIMING ARRAYS CONTAIN TIMING POINTS, E.G. 0 FOR T0, 1
',/,
    $' FOR T1, 2 FOR T2, ETC. ',/,
    $' THE VALVE ARRAYS CONTAIN BIT NUMBERS OF THE VALVES, A
MINUS ',/,
    $' SIGN INDICATES A CLOSE REACTION, A PLUS SIGN AN OPEN RE-',/,
    $' ACTION. ')
C H TEST
```

```
      WRITE(IPR,1450)((NTH(I),NVH(I)),I=1,NS(1))
1450 FORMAT(
    $'OSET UP FOR THE H TEST....ARRAYS NTH AND NVH ',/,
    $'OTIME POINT      BIT TO ACTIVATE ',/,
    $'                  - = CLOSE, + = OPEN ',//,
    $(5X,I2,T25,I4))
```

```
C SET UP FOR SCI TEST
      WRITE(IPR,1460)((NTSCI(I),NVSCI(I)),I=1,NS(2))
1460 FORMAT(
    $'1SET UP FOR THE SCI TEST....ARRAYS NTSCI AND NVSHI ',/,
    $'OTIME POINT      BIT TO ACTIVATE ',/,
    $'                  - = CLOSE, + = OPEN ',//,
    $(5X,I2,T25,I4))
```

```
C SET UP FOR SH TEST
      WRITE(IPR,1470)((NTSH(I),NVSH(I)),I=1,NS(3))
1470 FORMAT(
    $'OSET UP FOR THE SH TEST....ARRAYS NTSH AND NVSH ',/,
```

```
$'OTIME POINT      BIT TO ACTIVATE',/,
$'                  - = CLOSE, + = OPEN',//,
$(5X,I2,T25,I4))
C SET UP FOR SHI TEST
  WRITE(IPR,1480)((NTSHI(I),NVSHI(I)),I=1,NS(4))
1480 FORMAT(
  $'1SET UP FOR THE SHI TEST....ARRAYS NTSHI AND NVSHI',/,
  $'OTIME POINT      BIT TO ACTIVATE',/,
  $'                  - = CLOSE, + = OPEN',//,
  $(5X,I2,T25,I4))
C SET UP FOR SECI TEST
  WRITE(IPR,1490)((NTSECI(I),NVSECI(I)),I=1,NS(5))
1490 FORMAT(
  $'0SET UP FOR THE SECI TEST....ARRAYS NTSECI AND NVSECI',/,
  $'OTIME POINT      BIT TO ACTIVATE',/,
  $'                  - = CLOSE, + = OPEN',//,
  $(5X,I2,T25,I4))
  WRITE(IPR,1500)
1500 FORMAT(
  $'1FROM THESE TABLES WE CAN EXTRACT THE NUMBER OF TIME POINTS',/,
  $' FOR EACH TEST. THESE NUMBERS ARE GIVEN IN THE NT ARRAY.')
  WRITE(IPR,1510)(NT(I),I=1,5)
1510 FORMAT(
  $'0ARRAY          NUMBER OF TIME POINTS',//,
  $' NTH,NVH        ',I2,/,
  $' NTSCI,NVSCI    ',I2,/,
  $' NTSH,NVSH      ',I2,/,
  $' NTSHI,NVSHI    ',I2,/,
  $' NTSECI,NVSECI  ',I2)
  WRITE(IPR,1520)
1520 FORMAT(
  $'0TVL1,TVL2,TVL3,TVL4 AND TVL5 ARE DELAY TIMES. ',/,
  $'0TVL1 IS THE TIME BETWEEN T0 AND T1.',/,
  $'0TVL2 IS THE TIME BETWEEN :',/,
  $' - T1 AND T2   FOR THE H TEST',/,
  $' - T1 AND T2, T2 AND T3 ',/,
  $' TVL2 MULTIPLIED WITH THE SFCC IGNITION TIME GIVES THE TIME',/,
```

```
$' BETWEEN T4 AND T5 FOR THE SHI TEST AND BETWEEN T2 AND T3',/,
$' FOR THE SECI TEST.',/,
$'OTVL3 IS USED AS PURGE TIME (NITROGEN VALVE OPEN).',/,
$'OTVL4 IS THE TIME BETWEEN :',/,
$' - T5 AND T6 FOR THE SH TEST',/,
$' - T7 AND T8 FOR THE SHI TEST',/,
$' - T5 AND T6 FOR THE SECI TEST',/,
$'OTVL5 IS THE TIME BETWEEN T3 AND T4 FOR THE SH,SEH,SHI',/,
$' AND SEHI TEST.')
WRITE(IPR,1530)TVL1,TVL2,TVL3,TVL4,TVL5
1530 FORMAT(
$'OTVL1 = ',F10.5,' S',/,
$' TVL2 = ',F10.5,' S',/,
$' TVL3 = ',F10.5,' S',/,
$' TVL4 = ',F10.5,' S',/,
$' TVL5 = ',F10.5,' S')
C COMMON BLOCK BD6
WRITE(IPR,999)
WRITE(IPR,1600)
1600 FORMAT(
$'OCOMMON BLOCK BD6 IS DEFINED AS :',/,
$'OCOMMON/BD6/NST(6),NV(100)',/,
$'OTHIS BLOCK IS USED TO GET THE CORRECT SETTING OF THE MAN- ',/,
$' UAL VALVES DURING THE TEST')
WRITE(IPR,1610)
1610 FORMAT(
$'ONST GIVES THE STARTING ELEMENT IN NV FOR A PARTICULAR GAS',/,
$' LINE.',/,
$' THE VALUE OF THIS ELEMENT IS THE NUMBER OF ELEMENTS (STAR-',/,
$' TING AT THIS ELEMENT) FOR THAT PARTICULAR LINE.',/,
$' NV GIVES PER LINE THE REQUIRED SETTING OF THE MANUAL
VALUES.',/,
$' VALUES SUPERSEDED BY 900 ARE NOT STRICT (BUFFERVESSELS)',/,
$' AND VALUES SUPERSEDED BY 999 ARE STRICT')
WRITE(IPR,1620)((I,NST(I),NV(NST(I))),I=1,6)
1620 FORMAT(
$'OLINE STARTING ELEMENT IN ARRAY VALUE OF THIS',/,
```

```

$'          NV FOR THIS LINE          ELEMENT',//,
$(2X,I2,T25,I3,T48,I3))
  WRITE(IPR,1630)
1630 FORMAT(
  '$1ARRAY NV HAS THE FOLLOWING CONTENTS',/,
  '$OELEMENT CONTENTS  ELEMENT CONTENTS  ELEMENT CONTENTS',/)
  WRITE(IPR,1640)((I,NV(I)),I=1,100)
1640 FORMAT(
  $(3X,I3,7X,I3,7X,I3,7X,I3,7X,I3,7X,I3))
C FORCE NEW PAGE
  WRITE(IPR,1645)
1645 FORMAT(1H1)
C TYPE OUT SETTING PROCEDURE
  DO 130 I=1,6
C GET STARTING ELEMENT
  NO = NST(I)
C GET VALUE IN NV
  N1 = NV(NO)
C ZERO NSTEP
  NSTEP = 0
  WRITE(IPR,1650)I
1650 FORMAT(
  '$OSET UP PROCEDURE FOR LINE ',I2,' *****',/)
C LOOP OVER NUMBER OF OPERATIONS
  DO 140 J=1,N1
  IF(NV(J+NO).EQ.999) GOTO 160
  IF(NV(J+NO).EQ.900) GOTO 150
C PRINT OUT
  NSTEP = NSTEP + 1
  IF(NV(J+NO).LT.0) WRITE(IPR,1660)NSTEP,IABS(NV(J+NO)),AMK
  IF(NV(J+NO).GT.0) WRITE(IPR,1670)NSTEP,IABS(NV(J+NO)),AMK
1660 FORMAT(
  '$ STEP ',I2,' CLOSE VALVE ',I2,3X,A3,1X,'STRICT')
1670 FORMAT(
  '$ STEP ',I2,' OPEN VALVE ',I2,3X,A3,1X,'STRICT')
  GOTO 140
150 AMK = 'NOT'
```

```
GOTO 140
160 AMK = ' '
140 CONTINUE
130 CONTINUE
C COMMON BLOCK BD7
  WRITE(IPR,999)
  WRITE(IPR,1700)
1700 FORMAT(
  $'OCOMMON BLOCK BD7 IS DEFINED AS :',/,
  $'OCOMMON BLOCK/BD7/FLM1,FLM2,FLM3,FLM4,FLM5,FLM6,FLM7,IJPM(7)')
  WRITE(IPR,1710)
1710 FORMAT(
  $'OFLM1 MULTIPLIED WITH THE OXYGEN MASSFLOW GIVES THE H2
MASS-',/,
  $' FLOW.',/,
  $'OFLM2 IS THE MAXIMUM ALLOWABLE H2 MASSFLOW.',/,
  $'OFLM3 IS USED TO OBTAIN THE MASSFLOW FOR THE NON-SCMC O2',/,
  $' LINES FROM THE OXIDIZER MASSFLOW (FLM3 IS USED AS A MULTI-',/,
  $' PLIER).',/,
  $'OFLM4 MULTIPLIED WITH THE OXYGEN MASSFLOW GIVES THE CH4
MASS-',/,
  $' FLOW.',/,
  $'OFLM5 IS THE MAXIMUM ALLOWABLE SECONDARY PRESSURE.',/,
  $'OFLM6 IS THE MASSFLOW FOR THE O2(2) LINE IN CASE OF A',/,
  $' SH,SHI,SEH OR A SEHI TEST.',/,
  $'OFLM7 MULTIPLIED WITH THE ESTIMATED PRESSURE IN THE SFCC',/,
  $' MAY NOT EXCEED ANY SECONDARY PRESSURE.',/,
  $'OIJPM CONTAINS THE DEVICE TYPE(SEE ALSO BLOCK BD4).')
  WRITE(IPR,1720)FLM1,FLM2,FLM3,FLM4,FLM5,FLM6,FLM7
1720 FORMAT(
  $'OFLM1 = ',F10.5,/,
  $' FLM2 = ',F10.5,' KG/S',/,
  $' FLM3 = ',F10.5,/,
  $' FLM4 = ',F10.5,/,
  $' FLM5 = ',E15.5,' PASCAL',/,
  $' FLM6 = ',F10.5,' KG/S',/,
  $' FLM7 = ',F10.5)
```

```
WRITE(IPR,1730)((I,IJPM(I)),I=1,7)
1730 FORMAT(
  $'OLINE      DEVICE TYPE',//,
  $(2X,I2,T16,I2))
C COMMON BLOCK IJK
  WRITE(IPR,999)
  WRITE(IPR,1800)
1800 FORMAT(
  $'OCOMMON BLOCK IJK IS DEFINED AS :',/,
  $'OCOMMON/IJK/AIJK(21),BIJK(21),CIJK(3),DIJK(3)',/,
  $'OTHIS COMMON BLOCK IS USED TO CONVERT VOLTAGES TO ',/,
  $' TEMPERATURES OR PRESSURES OR TO CONVERT MASSFLOWS TO VOL-',/,
  $' TAGES')
  WRITE(IPR,1810)
1810. FORMAT(
  $'OAIJK AND BIJK WILL CONVERT VOLTAGES COMING FROM THE A/D',/,
  $' CONVERTERS ACCORDING TO :',/,
  $'OTEMPERATURE (IN KELVIN) OR PRESSURE (IN PASCAL) =',/,
  $'          VOLTAGE * AIJK + BIJK')
  WRITE(IPR,1820)((I-1,AIJK(I),BIJK(I)),I=1,21)
1820 FORMAT(
  $'OA/D CHANNEL          AIJK          BIJK',//,
  $(6X,I2,T20,E15.5,5X,E15.5))
  WRITE(IPR,1830)
1830 FORMAT(
  $'OCIJK AND DIJK WILL CONVERT MASSFLOWS TO VOLTAGES, WHICH
ARE',/,
  $' USED BY THE D/A CONVERTS, ACCORDING TO :',/,
  $'OVOLTAGE = MASSFLOW (IN KG/S) * CIJK + DIJK')
  WRITE(IPR,1840)((I-1,CIJK(I),DIJK(I)),I=1,3)
1840 FORMAT(
  $'OD/A CHANNEL          CIJK          DIJK',//,
  $(6X,I2,T20,E15.5,5X,E15.5))
C END OF PROGRAM
C GIVE MESSAGE ON TERMINAL IN CASE OF FILE
  IF(KPR.GT.0) WRITE(IOUT,2000)
2000 FORMAT(1H0,'FILE VBLOK.TXT CONTAINS THE OUTPUT OF THE',
```


\$ ' PROGRAM')

END

APPENDIX B

Voorbeeld van een print-out van alle gegevens,
die in de BLOCK DATA routine en het
SFCC programma worden gebruikt

```
*****  
****  
**** PROGRAM VBLOK ****  
**** SEKTIE RAKETVOORTSTUWING ****  
**** DATUM 30-JAN-84 ****  
*****
```

PROGRAM VBLOK, A PROGRAM TO PRINT OUT THE DATA STORED IN
THE BLOCK DATA OF THE SFCC PROGRAM
VERSION OF 22-12-83

COMMON BLOCK B01 IS DEFINED AS :

COMMON/B01/T0(4),C0(4),C1(4),C2(4),C3(4),C4(4).

THESE CONSTANTS ARE USED TO CALCULATE THE RELATIVE MASSFLOWS WHEN USING THE VITIATOR.

THE METHOD OF CALCULATION IS GIVEN BY R.G.VAN BRUGGEN IN SFCC PUBLICATION #1.

THE RELATIVE METHAAN-MASSFLOW GIVEN IN TABLE 4 OF THIS PUBLICATION IS APPROXIMATED BY A FOURTH ORDER POLYNOMIAL WITH CONSTANTS C0,C1,C2,C3 AND C4, IN THE TEMPERATURE T(E).

FOR EACH TEMPERATURE T(0) A POLYNOMIAL IS GIVEN.

TEMP	POLYNOMIAL CONSTANTS				
T0	C0	C1	C2	C3	C4
240.	-0.47189E+01	0.19588E-01	-0.32438E-06	0.28232E-08	-0.72344E-12
270.	-0.53445E+01	0.19812E-01	-0.96842E-06	0.34881E-08	-0.96837E-12
300.	-0.59138E+01	0.19672E-01	-0.82130E-06	0.34689E-08	-0.10095E-11
330.	-0.65173E+01	0.19789E-01	-0.13300E-05	0.39954E-08	-0.11824E-11

FOR THIS TABLE :

TEMPERATURE IN KELVIN.

COMMON BLOCK BD2 IS DEFINED AS :

COMMON/BD2/GM(7),V(18),VL(7),NPB(18),NBB(7).

THIS COMMON BLOCK IS USED TO CALCULATE THE ACTUAL BUFFER-VOLUMES.

GM CONTAINS THE MOLECULAIR MASSES FOR THE USED GASES,
NBB CONTAINS THE NUMBER OF BUFFERVESSELS PER USED LINE,
VL CONTAINS THE VOLUMES OF THE PIPES PER LINE.

GAS	MOLECULAIR MASS G/MOL	NUMBER OF BUFFERVESSELS	VOLUME OF PIPES M**3
AIR	28.96000	1	0.00001
O2(1)	32.00000	3	0.00001
O2(2)	32.00000	2	0.00001
O2(3)	32.00000	2	0.00001
H2	2.02000	4	0.00001
CH4(1)	16.04000	3	0.00001
CH4(2)	16.04000	3	0.00001

THE TOTAL NUMBER OF BUFFERVESSELS ON THIS LIST IS 18.
FOR EACH VESSEL INFORMATION ABOUT THE VOLUME AND THE NUMBER
OF THE INPUTBIT IS GIVEN IN RESP. V AND NPB.

BUFFERVESSEL	VOLUME M**3	INPUTBIT
1	3.0240	1
2	0.0500	3
3	0.0500	4
4	0.0500	5
5	0.0500	8
6	0.0500	9
7	0.0500	8
8	0.0500	9
9	0.0010	13
10	0.0010	14
11	0.0010	15
12	0.0010	16
13	0.0500	19
14	0.0500	20
15	0.0100	21
16	0.0500	19
17	0.0500	20
18	0.0100	21

WITH THE AID OF THE NBB,V AND NPB ARRAY WE CAN CALCULATE
THE ACTUAL BUFFERVOLUME.
THE TOTAL BUFFERVOLUME CAN BE CALCULATED FROM NBB AND V :

GAS	TOTAL BUFFER VOLUME M**3
AIR	3.02400
O2(1)	0.15000
O2(2)	0.10000
O2(3)	0.10000
H2	0.00400
CH4(1)	0.11000
CH4(2)	0.11000

COMMON BLOCK BD3 IS DEFINED AS :

COMMON/BD3/AMARK1(7),AMARK2(7),NPP(7),NPS(7),NTT(7).

AMARK1 AND AMARK2 CONTAIN INFORMATION ABOUT THE TYPE OF GAS,
NPP CONTAINS THE A/D CHANNEL NUMBER FOR THE PRIMARY PRES-
SURE PER LINE,
NPS CONTAINS THE A/D CHANNEL NUMBER FOR THE SECONDAY PRES-
SURE PER LINE,
NTT CONTAINS THE A/D CHANNEL NUMBER FOR THE TEMPERATURE PER
LINE.

LINE	GAS (AMARK1+AMARK2)	NPP	NPS	NTT
1	AIR	1	2	3
2	O2(1)	7	8	9
3	O2(2)	10	11	9
4	O2(3)	10	12	9
5	H2	13	14	9
6	CH4(1)	15	16	17
7	CH4(2)	18	19	17

COMMON BLOCK BD4 IS DEFINED AS :

COMMON/BD4/TNUL(7),PL(3,7),Z1(7),Z2(7),Z3(7),Y1(7),Y2(7).

THIS BLOCK CONTAINS INFORMATION ABOUT THE MASSFLOW DEVICES.

TNUL CONTAINS THE CALIBRATION TEMPERATURE (ONLY FOR SCMC DEVICES), THE CALCULATED MASSFLOW WILL BE CORRECTED WITH SQRT(TNUL/ACTUAL TEMPERATURE).

PL(2,J) CONTAINS THE MINIMUM ACCEPTABLE DIAFRAGM (IN MM), OR PENETRATION LENGTH (IN MM) OR NUMBER OF TURNS, PL(3,J) CONTAINS THE MAXIMUM ACCEPTABLE DIAFRAGM (IN MM), OR PENETRATION LENGTH (IN MM) OR NUMBER OF TURNS.

GAS	TNUL K	PL(2,J)	PL(3,J)
AIR	0.0	4.50	10.00
O2(1)	300.0	50.00	69.00
O2(2)	0.0	0.05	1.00
O2(3)	0.0	0.05	1.00
H2	0.0	0.00	1.00
CH4(1)	0.0	4.50	10.00
CH4(2)	0.0	0.00	1.00

Z1,Z2 AND Z3 ARE THE CONSTANTS FOR A SECOND ORDER POLYNOMIAL WHICH GIVES THE RELATION BETWEEN AT AND (DIAFRAGM/PENETRATION LENGTH) OR BETWEEN MASSFLOW/PRESSURE AND (NUMBER OF TURNS) :

AT OR MASSFLOW/PRESSURE = Z3 + Z2 * L + Z1 * L * L

LINE	TYPE DEVICE	GAS	Z3	Z2	Z1
1	1	AIR	0.00000	0.00000	0.78540
2	2	O2(1)	46.30000	-0.17707	-0.00591
3	3	O2(2)	-10.08000	205.78000	-76.03000
4	3	O2(3)	-10.08000	205.78000	-76.03000
5	3	H2	0.22460	1.70520	-0.64160
6	1	CH4(1)	0.00000	0.00000	0.78540
7	3	CH4(2)	-10.08000	205.78000	-76.03000

FOR THIS TABLE :

L IN MM, AT IN MM**2, MASSFLOW IN G/S, PRESSURE IN MPA.

DEVICE TYPE IS GIVEN IN COMMON BLOCK BD7 AND HAS THE FOLLOWING MEANING :

- TYPE 1 = DEVICE WITH FIXED DIAFRAGM
- TYPE 2 = SCMC DEVICE
- TYPE 3 = DEVICE WITH ADJUSTABLE NUMBER OF TURNS

Y1 AND Y2 ARE CONSTANTS IN THE RELATION BETWEEN MASSFLOW/AT AND PRESSURE FOR DEVICES WITH DIAFRAGMS AND PENETRATION-LENGTHS AND CONSTANTS IN THE RELATION BETWEEN MASSFLOW AND PRESSURE FOR DEVICES WITH NUMBER OF TURNS :

$$\text{MASSFLOW/AT OR MASSFLOW} = Y1 + Y2 * \text{PRESSURE}$$

LINE	TYPE DEVICE	GAS	Y1	Y2
1	1	AIR	0.27744E-08	0.44326E-03
2	2	O2(1)	0.24510E-08	0.34900E-04
3	3	O2(2)	0.10000E-08	0.00000E+00
4	3	O2(3)	0.10000E-08	0.00000E+00
5	3	H2	0.10000E-08	0.00000E+00
6	1	CH4(1)	0.27744E-08	0.44326E-03
7	3	CH4(2)	0.10000E-08	0.00000E+00

FOR THIS TABLE :

MASSFLOW IN KG/S, PRESSURE IN PASCAL, AT IN MM**2.

COMMON BLOCK BD5 IS DEFINED AS :

COMMON/BD5/NS(5),NTH(12),NTSCI(12),NTSH(22),NTSHI(32),
NTSECI(16),NT(5),NVH(12),NVSCI(12),NVSH(22),
NVSHI(32),NVSECI(16),TVL1,TVL2,TVL3,TVL4,TVL5

NS GIVES INFORMATION ABOUT THE LENGTH OF THE NTH/NVH,
NTSCI/NVSCI,NTSH/NVSH,NTSHI/NVSHI AND NTSECI/NVSECI ARRAYS
RESPECTIVELY.

ARRAY	NUMBER OF ELEMENTS
NTH,NVH	12
NTSCI,NVSCI	12
NTSH,NVSH	22
NTSHI,NVSHI	32
NTSECI,NVSECI	16

ALL THESE ARRAY ARE USED TO SET UP THE TIMING OF THE VALVES,
THE SECOND CHARACTER OF THE ARRAY NAME INDICATES WHETHER
THE ARRAY IS USED AS TIMING (T) ARRAY OR AS VALVE (V) ARRAY.
CHARACTERS 3 TILL END ARE USED TO INDICATE THE TYPE OF THE
EXPERIMENT.

THE TIMING ARRAYS CONTAIN TIMING POINTS, E.G. 0 FOR T0, 1
FOR T1, 2 FOR T2, ETC.

THE VALVE ARRAYS CONTAIN BIT NUMBERS OF THE VALVES, A MINUS
SIGN INDICATES A CLOSE REACTION, A PLUS SIGN AN OPEN RE-
ACTION.

SET UP FOR THE H TEST....ARRAYS NTH AND NVH

TIME POINT BIT TO ACTIVATE
 - = CLOSE, + = OPEN

1	98
1	100
2	103
2	104
2	119
3	-103
3	-119
4	-98
4	111
4	-100
5	-111
5	-104

SET UP FOR THE SCI TEST....ARRAYS NTSCI AND NVSHI

TIME POINT	BIT TO ACTIVATE
	- = CLOSE, + = OPEN
1	102
1	103
1	104
1	119
2	97
3	-102
3	-103
3	-119
4	-97
4	111
5	-111
5	-104

SET UP FOR THE SH TEST....ARRAYS NTSH AND NVSH

TIME POINT	BIT TO ACTIVATE
	- = CLOSE, + = OPEN
1	101
1	107
1	108
1	118
2	97
2	100
2	105
2	106
3	-101
3	-107
3	-118
4	98
5	-98
5	109
5	110
6	-97
6	-100
6	-105
7	-109
7	-110
7	106
7	-108

SET UP FOR THE SHI TEST....ARRAYS NTSHI AND NVSHI

TIME POINT BIT TO ACTIVATE
- = CLOSE, + = OPEN

1	101
1	107
1	108
1	118
2	97
2	100
2	105
2	106
3	-101
3	-107
3	-118
4	102
4	103
4	104
4	119
5	98
6	-102
6	-103
6	-119
7	-98
8	-97
8	-100
8	-105
8	109
8	110
8	111
9	-111
9	-110
9	-109
9	-106
9	-108
9	-104

SET UP FOR THE SECI TEST....ARRAYS NTSECI AND NVSECI

TIME POINT BIT TO ACTIVATE
- = CLOSE, + = OPEN

1	97
1	100
2	102
2	103
2	104
2	119
3	98
4	-102
4	-103
4	-119
5	-98
5	111
6	-97
6	-100
7	-111
7	-104

FROM THESE TABLES WE CAN EXTRACT THE NUMBER OF TIME POINTS FOR EACH TEST. THESE NUMBERS ARE GIVEN IN THE NT ARRAY.

ARRAY NUMBER OF TIME POINTS

NTH, NVH	6
NTSCI, NVSCI	6
NTSH, NVSH	8
NTSHI, NVSHI	10
NTSECI, NVSECI	8

TVL1, TVL2, TVL3, TVL4 AND TVL5 ARE DELAY TIMES.

TVL1 IS THE TIME BETWEEN T0 AND T1.

TVL2 IS THE TIME BETWEEN :

- T1 AND T2 FOR THE H TEST
- T1 AND T2, T2 AND T3

TVL2 MULTIPLIED WITH THE SFCC IGNITION TIME GIVES THE TIME BETWEEN T4 AND T5 FOR THE SHI TEST AND BETWEEN T2 AND T3 FOR THE SECI TEST.

TVL3 IS USED AS PURGE TIME (NITROGEN VALVE OPEN).

TVL4 IS THE TIME BETWEEN :

- T5 AND T6 FOR THE SH TEST
- T7 AND T8 FOR THE SHI TEST
- T5 AND T6 FOR THE SECI TEST

TVL5 IS THE TIME BETWEEN T3 AND T4 FOR THE SH, SEH, SHI AND SEHI TEST.

TVL1	=	1.00000	S
TVL2	=	0.50000	S
TVL3	=	15.00000	S
TVL4	=	2.50000	S
TVL5	=	30.00000	S

COMMON BLOCK BD6 IS DEFINED AS :

COMMON/BD6/NST(6),NV(100)

THIS BLOCK IS USED TO GET THE CORRECT SETTING OF THE MANUAL VALVES DURING THE TEST

NST GIVES THE STARTING ELEMENT IN NV FOR A PARTICULAR GAS LINE.

THE VALUE OF THIS ELEMENT IS THE NUMBER OF ELEMENTS (STARTING AT THIS ELEMENT) FOR THAT PARTICULAR LINE.

NV GIVES PER LINE THE REQUIRED SETTING OF THE MANUAL VALVES. VALUES SUPERSEDED BY 900 ARE NOT STRICT (BUFFERVESSELS), AND VALUES SUPERSEDED BY 999 ARE STRICT

LINE	STARTING ELEMENT IN ARRAY NV FOR THIS LINE	VALUE OF THIS ELEMENT
1	1	4
2	6	12
3	19	11
4	31	12
5	44	13
6	58	12

ARRAY NV HAS THE FOLLOWING CONTENTS

ELEMENT	CONTENTS	ELEMENT	CONTENTS	ELEMENT	CONTENTS
1	4	2	999	3	-1
4	999	5	1	6	12
7	999	8	-6	9	999
10	2	11	900	12	3
13	4	14	5	15	999
16	-2	17	999	18	6
19	11	20	999	21	-10
22	999	23	7	24	900
25	8	26	9	27	999
28	-7	29	999	30	10
31	12	32	999	33	-10
34	999	35	7	36	900
37	8	38	9	39	999
40	-7	41	999	42	10
43	11	44	13	45	999
46	-17	47	999	48	12
49	900	50	13	51	14
52	15	53	16	54	999
55	-12	56	999	57	17
58	12	59	999	60	-22
61	999	62	18	63	900
64	19	65	20	66	21
67	999	68	-18	69	999
70	22	71	999	72	999
73	999	74	999	75	999
76	999	77	999	78	999
79	999	80	999	81	999
82	999	83	999	84	999
85	999	86	999	87	999
88	999	89	999	90	999
91	999	92	999	93	999
94	999	95	999	96	999
97	999	98	999	99	999
100	999				

SET UP PROCEDURE FOR LINE 1 *****

STEP 1 CLOSE VALVE 1 STRICT
STEP 2 OPEN VALVE 1 STRICT

SET UP PROCEDURE FOR LINE 2 *****

STEP 1 CLOSE VALVE 6 STRICT
STEP 2 OPEN VALVE 2 STRICT
STEP 3 OPEN VALVE 3 NOT STRICT
STEP 4 OPEN VALVE 4 NOT STRICT
STEP 5 OPEN VALVE 5 NOT STRICT
STEP 6 CLOSE VALVE 2 STRICT
STEP 7 OPEN VALVE 6 STRICT

SET UP PROCEDURE FOR LINE 3 *****

STEP 1 CLOSE VALVE 10 STRICT
STEP 2 OPEN VALVE 7 STRICT
STEP 3 OPEN VALVE 8 NOT STRICT
STEP 4 OPEN VALVE 9 NOT STRICT
STEP 5 CLOSE VALVE 7 STRICT
STEP 6 OPEN VALVE 10 STRICT

SET UP PROCEDURE FOR LINE 4 *****

STEP 1 CLOSE VALVE 10 STRICT
STEP 2 OPEN VALVE 7 STRICT
STEP 3 OPEN VALVE 8 NOT STRICT
STEP 4 OPEN VALVE 9 NOT STRICT
STEP 5 CLOSE VALVE 7 STRICT
STEP 6 OPEN VALVE 10 STRICT
STEP 7 OPEN VALVE 11 STRICT

SET UP PROCEDURE FOR LINE 5 *****

STEP 1 CLOSE VALVE 17 STRICT
STEP 2 OPEN VALVE 12 STRICT
STEP 3 OPEN VALVE 13 NOT STRICT
STEP 4 OPEN VALVE 14 NOT STRICT
STEP 5 OPEN VALVE 15 NOT STRICT
STEP 6 OPEN VALVE 16 NOT STRICT
STEP 7 CLOSE VALVE 12 STRICT
STEP 8 OPEN VALVE 17 STRICT

SET UP PROCEDURE FOR LINE 6 *****

STEP 1 CLOSE VALVE 22 STRICT
STEP 2 OPEN VALVE 18 STRICT
STEP 3 OPEN VALVE 19 NOT STRICT
STEP 4 OPEN VALVE 20 NOT STRICT
STEP 5 OPEN VALVE 21 NOT STRICT
STEP 6 CLOSE VALVE 18 STRICT
STEP 7 OPEN VALVE 22 STRICT

COMMON BLOCK BD7 IS DEFINED AS :

COMMON BLOCK/BD7/FLM1,FLM2,FLM3,FLM4,FLM5,FLM6,FLM7,IJPM(7)

FLM1 MULTIPLIED WITH THE OXYGEN MASSFLOW GIVES THE H2 MASSFLOW.

FLM2 IS THE MAXIMUM ALLOWABLE H2 MASSFLOW.

FLM3 IS USED TO OBTAIN THE MASSFLOW FOR THE NON-SCMC O2 LINES FROM THE OXIDIZER MASSFLOW (FLM3 IS USED AS A MULTIPLIER).

FLM4 MULTIPLIED WITH THE OXYGEN MASSFLOW GIVES THE CH4 MASSFLOW.

FLM5 IS THE MAXIMUM ALLOWABLE SECONDARY PRESSURE.

FLM6 IS THE MASSFLOW FOR THE O2(2) LINE IN CASE OF A SH,SHI,SEH OR A SEHI TEST.

FLM7 MULTIPLIED WITH THE ESTIMATED PRESSURE IN THE SFCC MAY NOT EXCEED ANY SECONDARY PRESSURE.

IJPM CONTAINS THE DEVICE TYPE(SEE ALSO BLOCK BD4).

- FLM1 = 0.12500
- FLM2 = 0.01200 KG/S
- FLM3 = 0.05000
- FLM4 = 0.16700
- FLM5 = 0.80000E+07 PASCAL
- FLM6 = 0.01500 KG/S
- FLM7 = 2.00000

LINE DEVICE TYPE

1	1
2	2
3	3
4	3
5	3
6	1
7	3

COMMON BLOCK IJK IS DEFINED AS :

COMMON/IJK/AIJK(21),BIJK(21),CIJK(3),DIJK(3)

THIS COMMON BLOCK IS USED TO CONVERT VOLTAGES TO TEMPERATURES OR PRESSURES OR TO CONVERT MASSFLOWS TO VOLTAGES

AIJK AND BIJK WILL CONVERT VOLTAGES COMING FROM THE A/D CONVERTERS ACCORDING TO :

TEMPERATURE (IN KELVIN) OR PRESSURE (IN PASCAL) = VOLTAGE * AIJK + BIJK

A/D CHANNEL	AIJK	BIJK
0	0.40000E+07	0.00000E+00
1	0.20000E+07	0.00000E+00
2	0.80000E+02	0.00000E+00
3	0.20000E+07	0.00000E+00
4	0.80000E+02	0.00000E+00
5	0.80000E+02	0.00000E+00
6	0.40000E+07	0.00000E+00
7	0.20000E+07	0.00000E+00
8	0.80000E+02	0.00000E+00
9	0.40000E+07	0.00000E+00
10	0.20000E+07	0.00000E+00
11	0.20000E+07	0.00000E+00
12	0.40000E+07	0.00000E+00
13	0.20000E+07	0.00000E+00
14	0.40000E+07	0.00000E+00
15	0.20000E+07	0.00000E+00
16	0.80000E+02	0.00000E+00
17	0.40000E+07	0.00000E+00
18	0.20000E+07	0.00000E+00
19	0.20000E+07	0.00000E+00
20	0.80000E+02	0.00000E+00

CIJK AND DIJK WILL CONVERT MASSFLOWS TO VOLTAGES, WHICH ARE USED BY THE D/A CONVERTS, ACCORDING TO :

VOLTAGE = MASSFLOW (IN KG/S) * CIJK + DIJK

D/A CHANNEL	CIJK	DIJK
0	0.25000E+01	0.00000E+00
1	0.25000E+01	0.00000E+00
2	0.25000E+01	0.00000E+00

Rapport 430



60141080496