

①②

DEMANDE DE BREVET D'INVENTION

A1

②② Date de dépôt : 07.07.98.

③⑦ Priorité : 07.07.97 DE 19728971.

④③ Date de mise à la disposition du public de la
demande : 08.01.99 Bulletin 99/01.

⑤⑥ Liste des documents cités dans le rapport de
recherche préliminaire : *Ce dernier n'a pas été
établi à la date de publication de la demande.*

⑥⑦ Références à d'autres documents nationaux
apparentés :

⑦① Demandeur(s) : ROBERT BOSCH GMBH GESELLS-
CHAFT MIT BESCHRANKTER HAFTUNG — DE.

⑦② Inventeur(s) : SEE OLE.

⑦③ Titulaire(s) :

⑦④ Mandataire(s) : CABINET HERRBURGER.

⑤④ DISPOSITIF ET PROCEDE DE TRAITEMENT DE DONNEES.

⑤⑦ Dispositif et procédé de traitement de données com-
prenant une première installation pour un premier traitement
utilisant une ou plusieurs variables globales avec un proces-
seur et une seconde installation pour un second traitement
avec le processeur. Au début on établit une copie locale
sous la forme d'une variable globale d'au moins une varia-
ble globale utilisée par la première installation et qui peut
être interrompue lors de la copie; un affichage d'état lisible
par la première installation est mis à l'état par la seconde
installation au cours de la copie pour afficher la procédure
de copie en cours. La première installation vérifie avant une
quelconque interruption de la seconde installation, si l'affi-
chage d'état est mis et, dans l'affirmative, au moins une co-
pie locale en forme de variables globales est mise à la
valeur actuelle.

FR 2 765 700 - A1



Etat de la technique

La présente invention concerne un dispositif et un procédé de traitement de données comprenant :

- une première installation pour exécuter un premier traitement de données en appliquant une ou plusieurs variables globales à l'aide d'un processeur,
- une seconde installation pour l'exécution d'un second traitement de données à l'aide d'un processeur conçu pour qu'au début on établisse au moins une copie locale sous la forme d'une variable globale d'au moins l'une des variables globales utilisées par le premier processeur et lors de la copie, le premier processeur peut l'interrompre,
- une installation d'affichage d'état mise à l'état par la seconde installation lors de l'opération de copie et qui peut être lue par la première installation, pour afficher l'opération de copie en cours.

Bien que l'on puisse utiliser n'importe quel dispositif de traitement de données, la présente invention et le problème à la base de l'invention se rapportent à un dispositif de traitement de données embarqué dans un véhicule automobile servant à mettre en réseau les différents appareils de commande comportant des capteurs et des actionneurs comme par exemple un système de bus travaillant en temps réel ou encore appelé contrôleur de réseau de zone « Contrôler Area Network » (CAN).

On considère un appareil de commande servant à fixer un paramètre du moteur comme par exemple la quantité de carburant à injecter. Différents programmes ou applications du système de fonctionnement se déroulent dans cet appareil de commande qui utilise le processeur en temps partagé. Le temps de calcul du processeur est réparti entre les différents programmes. A chaque tâche est attribuée de manière cyclique ou selon certaines priorités, une partie de la puissance du calculateur. La commutation entre les différentes tâches se fait en général à une vitesse tellement élevée que les interruptions et les reprises ne sont pas perceptibles de l'extérieur. Toutes les données utilisées par plusieurs tâches sont définies comme des variables globales.

Lorsqu'on calcule dans une tâche A les données qui sont traitées ensuite dans une tâche B, il faut garantir la consistance ou cohérence des données si la tâche B peut être interrompue à tout instant par la tâche A. En d'autres
5 termes, avant la fin de la tâche B, il ne faut pas que son jeu de données actuel soit modifié, bien que la tâche A fournisse déjà de nouvelles données après l'interruption.

Pour expliciter de manière plus précise le problème, on considère le cas suivant de la régulation de la vitesse de rotation (régulation du régime) dans un appareil de
10 commande du moteur.

L'appareil de commande du moteur calcule la valeur de réglage du régulateur de vitesse de rotation. L'appareil de commande du moteur reçoit la valeur de consigne
15 de la vitesse de rotation à réguler ainsi que le mode de fonctionnement, c'est-à-dire le comportement de régulation (comme par exemple la partie proportionnelle, la partie différentielle ou la partie intégrale, etc...), par le bus CAN, à partir de l'appareil de commande du véhicule à intervalles
20 de 10 ms. Le calcul des paramètres de régulation (jeu de paramètres de régulation), c'est-à-dire du jeu de paramètres destiné au module de régulation, est reçu par contre dans l'appareil de commande du moteur toutes les 50 ms. Ainsi, il est clair que l'appareil de commande du moteur n'aura pas à
25 recevoir de nouvelles données de l'appareil de commande du véhicule jusqu'à la fin de sa durée de traitement respective de 50 ms bien que l'appareil de commande du véhicule fournisse de nouvelles données en principe toutes les 10 ms.

Actuellement, selon l'état de la technique, on
30 dispose de trois jeux garantissant la consistance des données.

Le premier jeu prévoit l'application de copies locales des données qui sont calculées avec la plus grande priorité, au début de la tâche à priorité la plus faible.

35 Un inconvénient de cette procédure est de ne plus pouvoir garantir la cohérence des données si la tâche de priorité plus faible est interrompue, pendant l'opération de copie proprement dite, par une tâche de priorité plus élevée.

En effet, dans ce cas, dans la poursuite de la tâche de priorité plus faible, à la fin de l'interruption, les copies déjà appliquées ne sont plus restaurées mais l'opération de copie reprend à l'endroit de l'interruption. C'est pourquoi, après
5 la fin de l'opération de copie, il y aura incohérence entre les données copiées en premier lieu et celles copiées en en second lieu.

Le second jeu prévoit l'application de copies locales de données comme pour le premier jeu. Pendant
10 l'opération de copie, on bloque toutefois toute interruption de sorte que la tâche de priorité la plus élevée ne peut plus « perturber » la tâche de priorité plus faible.

Cela présente néanmoins l'inconvénient que le temps de blocage pour la copie de quantités importantes de
15 données comme par exemple pour des structures, soit dépassé ce qui peut conduire à un redémarrage du système de fonctionnement. Ce comportement n'est pas souhaité car il bloque l'ensemble du système, comme par exemple les appareils de commande du moteur, pour des ratés à l'injection du carburant, ou même cela peut conduire à l'arrêt du moteur.
20

Le troisième jeu prévoit l'application d'une mémoire partagée. Cela constitue dans la mémoire active gérée par le système de fonctionnement une zone dans laquelle sont inscrites en principe toutes les données de programme ou dont
25 on peut extraire en principe toutes les données de programme si le système de fonctionnement le permet. Pour garantir la consistance des données dans le cas envisagé, une tâche de priorité élevée ne peut être enregistrée de nouveau dans la mémoire partagée que si la tâche de priorité inférieure a
30 terminé ses copies locales. Cela signifie que, pendant la réalisation des copies locales par la tâche de priorité basse, l'accès à la mémoire partagée est interdit à la tâche de priorité élevée.

Cela présente l'inconvénient qu'il faut gérer la
35 mémoire partagée par le système de fonctionnement ce qui aboutit à une mise en oeuvre de moyens plus importants, c'est-à-dire à une intervention plus importante dans le système de fonctionnement.

Comme inconvénient des solutions connues évoquées ci-dessus, on a également le fait que ces solutions ne présentent pas de sécurité suffisante et/ou nécessitent la mise en oeuvre de moyens trop importants.

5 **Avantages de l'invention**

La présente invention a pour but de remédier à ces inconvénients et concerne à cet effet, un dispositif de traitement de données du type défini ci-dessus caractérisé en ce que la première installation est conçue pour qu'avant la fin d'une
10 interruption respective de la seconde installation, elle vérifie que l'installation d'affichage d'état soit mise à l'état et que dans l'affirmative, au moins une copie sous forme de variables globales soit surscrite avec la valeur actuelle de ces variables.

15 L'invention concerne également un procédé de traitement de données caractérisé en ce qu' :

- on exécute un premier traitement de données en utilisant une ou plusieurs variables globales à l'aide d'une installation de processeur,
- 20 - on exécute un second traitement de données à l'aide de l'installation de processeur qui établit initialement au moins une copie locale sous la forme d'une variable globale d'au moins une variable globale utilisée par la première installation et qui, lors de la copie, peut être interrom-
- 25 pue par le premier traitement de données,
- on affiche l'opération de copie en cours du second traitement de données, d'une manière lisible par le premier traitement de données et,
- on contrôle dans le premier traitement de données, si
30 l'affichage est mis à l'état, avant la fin d'une interruption respective du second traitement de données et dans l'affirmative, le dépassement d'au moins une copie locale sous la forme de variables locales avec leur valeur actuelle.

35 Le dispositif et le procédé de traitement de données selon l'invention offrent l'avantage de permettre d'assurer une bonne cohérence des données avec des moyens

simples, c'est-à-dire sans la mise en oeuvre de matériel de programme important.

La cohérence des données est également assurée par des données qui sont en relation étroite et ainsi ne peuvent être traitées que dans les mêmes conditions, c'est-à-dire des données qui ne doivent pas provenir de deux cycles différents du premier traitement de données.

Les problèmes liés à l'interdiction d'interruption sont éliminés et l'accès au système est faible et se fait localement et non pas sur le plan du système de fonctionnement.

La caractéristique générale à la base de la présente invention est celle consistant à ce qu'une première tâche ou une première application de priorité élevée fixe certaines variables globales utilisées dans une seconde tâche de priorité plus faible. Pendant une opération de copie de données, dans la seconde tâche, on pose un drapeau global ou un affichage d'état lisible par une première tâche. En cas d'interruption de cette opération de copie de données par la première tâche, avant la fin de l'interruption, on vérifie l'état du drapeau dans la première tâche et, dans le cas où celui-ci est mis, cette information est considérée comme une indication d'une modification possible de données déjà lues. Ainsi, la première tâche ne surscrit que les variables modifiées ou toutes les variables locales déjà utilisées par la seconde tâche en leur surscrivant la valeur actuelle, variables qui sont déjà définies comme variables globales à cet effet. La réalisation de copies locales se fait au début de la seconde tâche de priorité faible, c'est-à-dire avant toute poursuite de traitement des données à copier. La surscription des copies locales se fait le cas échéant à la fin de la première tâche de priorité plus élevée, c'est-à-dire au plus tôt après que toutes les modifications des variables globales utilisées par la seconde tâche aient été terminées.

Suivant d'autres caractéristiques avantageuses de l'invention :

- on attribue une première priorité élevée à la première installation pour l'utilisation de l'installation de processeur et,
- on associe une seconde priorité faible à la seconde installation pour l'utilisation de l'installation de processeur,
- 5 - on associe une première durée courte à la première installation pour l'utilisation de l'installation de processeur et,
- on associe une seconde durée longue à la seconde installation pour l'utilisation de l'installation de processeur,
- 10 - on poursuit avec le second traitement de données à la fin d'une interruption respective à l'endroit de l'interruption,
- on détermine dans le premier traitement de données les copies locales utilisées par la seconde installation.
- 15

Selon un développement préférentiel, la première installation présente une première priorité élevée et la seconde installation une seconde priorité faible pour l'utilisation du processeur.

- 20 Selon un autre développement préférentiel, une première durée courte est associée à la première installation et une seconde durée longue est associée à la seconde installation pour l'utilisation du processeur.

- 25 Selon un autre développement préférentiel, la seconde installation est conçue pour qu'à la fin d'une interruption quelconque, elle continue à l'endroit de l'interruption. Cela est avantageux pour perdre le moins de temps possible pour l'exécution du second traitement de données.

- 30 Selon un autre développement préférentiel, la première installation comporte une installation de détermination qui sert à déterminer la copie locale ou les copies locales servant à la seconde installation. Cela présente l'avantage de n'avoir à remettre à jour que certaines variables modifiées fixées par le premier traitement de données ou
- 35 utilisées par le second traitement de données.

Dessins

Des exemples de réalisation de l'invention sont représentés dans les dessins et seront décrits ci-après de manière plus détaillée. Ainsi :

- 5 - la figure 1 montre un chronogramme d'une première tâche A et d'une seconde tâche B, si la première tâche A interrompt la seconde tâche B après une copie, selon un premier exemple de réalisation de l'invention,
- la figure 2 montre un chronogramme d'une première tâche A et d'une seconde tâche B lorsque la première tâche A interrompt la seconde tâche B lors d'une copie selon le premier exemple de réalisation de l'invention,
- la figure 3 est une vue schématique des étapes de traitement de données dans une première tâche A selon un exemple de réalisation de l'invention,
- 15 - la figure 4 est une vue schématique des étapes de traitement de données dans une seconde tâche selon un second exemple de réalisation de l'invention.

Description des exemples de réalisation

20 La figure 1 montre un chronogramme d'une première tâche A et d'une seconde tâche B lorsque la première tâche A interrompt la seconde tâche B après l'opération de copie, selon un premier exemple de réalisation de l'invention.

A la figure 1, les références t_1 , t_2 , t_3 , t_4 , t_5 désignent cinq instants successifs portés sur l'axe vertical des temps. Les références A1, A2, A3, A4 désignent des blocs de programme respectifs d'une tâche ou d'une application A, ayant une première priorité élevée. Les références B1, B2, B3, B4, B5 désignent des blocs de programme respectifs d'une tâche ou d'un programme d'application B ayant une seconde priorité plus faible. L'expression « bloc de programme » représente une partie de l'ensemble du programme.

Les deux tâches A et B sont traitées selon le programme de décalage de temps en tenant compte de leur priorité dans la même installation de processeur ou plus simplement processeur. En particulier, la tâche A calcule les variables V1 et V2 nécessaires à la tâche B. Pour cela il faut veiller à ce que la tâche B ne puisse utiliser qu'une

paire de données cohérentes pour V1 et V2 pendant l'exécution pour des raisons de cohérence des données.

Le traitement commence à l'instant t_1 avec le bloc de programme A1 de la tâche A qui contient entre autres
5 le calcul des données des variables globales V1 et V2. L'expression « variables globales » désigne dans la présente description le fait qu'au moins également la tâche B accède à de telles variables.

A l'instant t_2 , la tranche d'exécution de la tâche A se termine et la tâche B commence par le bloc de programme B1 ; dans celui-ci le drapeau F est mis à l'état
10 logique "1" ; ce drapeau F représente l'installation d'affichage d'état selon l'invention. Le drapeau F montre ainsi le début et le déroulement de l'opération de copie.

15 Ensuite, dans la tâche B, selon les blocs de programme B2 et B3, on effectue les copies locales des variables V1 et V2, à savoir GLK1 comme copie de V1 dans B2 et GLK2 comme copie de V2 dans B3.

Il est expressément indiqué que les variables
20 GLK1, GLK2 et F sont également des variables globales, c'est-à-dire des variables auxquelles peut accéder également la tâche A.

Dans le bloc de programme B4, le drapeau F est mis à l'état logique « 0 » pour indiquer la fin de
25 l'opération de copie.

A l'instant t_3 , la tranche d'exécution de la tâche B est terminée et la tâche A redémarre avec le bloc de programme A2 qui contient entre autres le nouveau calcul des données des variables globales V1 et V2.

30 Dans le bloc de programme suivant A3 qui arrive dans l'ordre après tous les nouveaux calculs des variables V1, V2 nécessaires à la tâche B, la tâche A vérifie si le drapeau F a été ou non mis à l'état. Comme dans le cas présent, l'interruption de la tâche s'est produite après
35 l'opération de copie et le drapeau F est à l'état logique « 0 ».

Comme la tâche A est informée que la tâche B est en possession des copies correspondantes nécessaires GLK1,

GLK2, et que ces copies n'ont pas été utilisées par la tâche A, aucune autre étape n'est nécessaire et la tâche A se termine avec le bloc de programme A4 à l'instant t_4 .

5 Puis la tâche B se poursuit avec le bloc de programme B5 qui termine la tâche B ou se poursuit jusqu'à l'interruption suivante par la tâche A. Toutefois, ni la première interruption décrite ni quelque'autres interruptions ne mettent en danger la cohérence des données dans la tâche B.

10 La figure 2 montre un chronogramme pour une première tâche A et une seconde tâche B, lorsque la première tâche A interrompt la seconde tâche B lors de l'opération de copie, selon le premier exemple de réalisation de l'invention.

15 A la figure 2, les références t_1 , t_2 , t_3' , t_4' , t_5 désignent cinq instants successifs représentés sur l'axe vertical des temps. Les références A1, A2, A3, A4' désignent des blocs de programme respectifs d'une tâche ou d'un programme d'application A ; les référence B1, B2, B3, B4, B5 désignent des blocs de programme respectifs des tâches ou
20 programme d'application B. Les références soulignées correspondent aux parties déjà décrites à la figure 1.

Le traitement commence comme ci-dessus à l'instant t_1 avec le bloc de programme A1 de la tâche A qui contient entre autres le calcul des données des variables
25 globales V1, V2.

A l'instant t_2 , le temps alloué à la tâche A est terminé et la tâche B commence par le bloc de programme B1 dans lequel le drapeau F est mis à l'état logique « 1 » ; cela affiche le début et le déroulement de l'opération de copie.
30

Puis, dans la tâche B, par le bloc de programme B2, on effectue une copie locale GLK1 des variables V1. A l'instant t_3' , on interrompt la tranche d'exécution de la tâche B par la tâche A ; la tâche A se poursuit par le bloc de programme A2 qui contient entre autres le nouveau calcul des
35 données des variables globales V1, V2.

Dans le bloc de programme suivant A3, qui arrive dans l'ordre, après tous les nouveaux calculs des variables

V1, V2 utilisés par la tâche B, la tâche A vérifie que le drapeau F a été ou non mis à l'état. Comme dans le cas présent l'interruption de la tâche B s'est effectuée pendant l'opération de copie et le drapeau F est à l'état logique
5 « 1 ».

La tâche A est ainsi informée de ce que la tâche B n'est pas en possession des copies cohérentes nécessaires GLK1, GLK2 et que d'autres étapes sont nécessaires pour assurer la cohérence des données.

10 Pour cela, la tâche A poursuit le bloc de programme A4' dans lequel s'exécute la copie de la tâche B, c'est-à-dire que dans GLK1 on a une copie de V1 et dans GLK2 une copie de V2.

Il convient de remarquer de manière générale que
15 la tâche A n'aura en principe qu'à renouveler les copies dont la variable correspondante s'est modifiée. Cela peut servir à des gains de temps pour des quantités importantes de données.

A l'instant t_4 , la tranche d'exécution de la tâche A est terminée et celle-ci se termine.

20 Puis la tâche B se poursuit à l'endroit où elle a été précédemment interrompue, et réalise dans le bloc de programme B3 une copie locale GLK2 des variables V2. Puis, dans le bloc de programme B4, on met le drapeau F à l'état logique « 0 » pour indiquer la fin de l'opération de copie.

25 Puis dans la tâche B on poursuit avec le bloc de programme B5 qui termine la tâche B ou se déroule jusqu'à l'interruption suivante par la tâche A. Ni la première interruption décrite ni une quelconque interruption ne mettant en danger la cohérence des données dans la tâche B. En particulier, toute autre interruption de la tâche B par la tâche A
30 conduit à la situation décrite en relation avec la figure 1.

La figure 3 montre un schéma des étapes de traitement de données dans une première tâche A selon le second exemple de réalisation de l'invention.

35 La tâche A occupe une tranche d'exécution ou de temps fixe de 10 ms. Dans les étapes S1 et S2, on calcule les données utilisées pour les variables globales V1, V2 uti-

lisées. Dans la zone en tirets, il y a n'importe quelle autre étape de traitement de données.

A la fin de la tâche A dans les étapes S4-S7, si nécessaire, c'est-à-dire si le drapeau F est à l'état logique « 1 », on répète l'opération de copie de V1 dans GLK1 et de V2 dans GLK2 pour la tâche B.

La figure 4 est une vue schématique des étapes de traitement de données dans une seconde tâche B selon le second exemple de réalisation de l'invention.

La tâche B occupe une tranche d'exécution fixe de 50 ms. Dans les étapes S1'-S4', on copie les variables globales V1, V2 utilisées par la tâche B dans les variables globales GLK1, GLK2. Ce n'est que pendant cette opération de copie, que le drapeau F est à l'état logique « 1 » pour afficher, lors d'une interruption par la tâche A, la nécessité de mesure pour éviter l'incohérence de données. Dans la zone en traits interrompus, il y a n'importe quelle autre étape de traitement de données.

A la fin de la tâche B, dans l'étape S5', on poursuit le traitement des copies locales GLK1 et/ou GLK2 ; la présente invention garantit alors que, pour le déroulement de la tâche B, on calcule toujours une seule et même paire de variables cohérentes.

Bien que la présente invention ait été décrite ci-dessus dans le cas d'un exemple de réalisation préférentiel, elle n'est pas limitée à cet exemple et peut être modifiée de manière très différente.

En particulier, les deux installations pour effectuer un traitement des données ainsi que l'installation d'affichage d'état, peuvent être réalisées non seulement sous la forme de programmes mais également sous la forme de circuits câblés.

En outre, dans le présent exemple de réalisation, les tâches A et B ont utilisé en commun deux variables V1 et V2. L'invention est toutefois applicable à un nombre quelconque de variables utilisées en commun. L'expression « variables » désigne ainsi de manière générale une grandeur variable.

En outre, la présente invention n'est pas seulement applicable pour accorder entre elles deux tâches avec un drapeau, mais plusieurs tâches qui utilisent chaque fois un drapeau par paire.

5 L'association de différents intervalles de temps à différentes tâches n'est pas indispensable. Une tâche de 50 ms est appelée à des intervalles de 50 ms ; une tâche de 10 ms est appelée à des intervalles de 10 ms. Cela signifie simplement qu'une tâche de 50 ms ne peut durer au maximum que
10 50 ms et qu'une tâche de 10 ms peut durer au maximum 10 ms pour que la tâche respective ne se répète pas d'elle même et déclenche ainsi un redémarrage du système de fonctionnement. Mais cela ne signifie pas que la durée doit être égale à 50 ms ou 10 ms.

NOMENCLATURE

$t_1, t_2, t_3, t_4, t_5,$ t_3', t_4'	Instantants
A1 - A4 - A4' B1 - B5	Blocs de programme
V1, V2	Variables globales
GLK1, GLK2	Variables globales pour les copies locales
F	Drapeau
S1 - S7	Etapes de programme pour la tâche A
S1' - S5'	Etapes de programme pour la tâche A

R E V E N D I C A T I O N S

1°) Dispositif de traitement de données comprenant :

- une première installation pour exécuter un premier traitement de données en appliquant une ou plusieurs variables globales à l'aide d'un processeur,
 - une seconde installation pour l'exécution d'un second traitement de données à l'aide d'un processeur conçu pour qu'au début on établisse au moins une copie locale sous la forme d'une variable globale d'au moins l'une des variables globales utilisées par le premier processeur et, lors de la copie, le premier processeur peut l'interrompre,
 - une installation d'affichage d'état mise à l'état par la seconde installation lors de l'opération de copie et qui peut être lue par la première installation, pour afficher l'opération de copie en cours,
- caractérisé en ce que
- la première installation est conçue pour qu'avant la fin d'une interruption respective de la seconde installation, elle vérifie que l'installation d'affichage d'état soit mise à l'état et que, dans l'affirmative, au moins une copie sous forme de variables globales soit surscrite avec la valeur actuelle de ces variables.

2°) Dispositif selon la revendication 1,

caractérisé en ce qu'

à la première installation est associée une première priorité élevée et à la seconde installation est associée une seconde priorité faible pour l'utilisation de l'installation de processeur.

3°) Dispositif selon la revendication 1 ou 2,

caractérisé en ce qu'

à la première installation est associée une première durée courte et à la seconde installation est associée une seconde durée longue pour l'utilisation de l'installation de processeur.

4°) Dispositif selon l'une quelconque des revendications précédentes,

caractérisé en ce que

la seconde installation est conçue pour qu'à la fin d'une interruption respective, elle poursuive à l'endroit de l'interruption.

5°) Dispositif selon l'une quelconque des revendications précédentes,

caractérisé en ce que

la première installation comporte une installation pour déterminer la copie locale ou les copies locales utilisées par la seconde installation.

6°) Procédé de traitement de données pour la mise en oeuvre du dispositif selon l'une quelconque des revendications 1 à 5,

caractérisé par les étapes suivantes :

- on exécute un premier traitement de données en utilisant une ou plusieurs variables globales à l'aide d'une installation de processeur,
- on exécute un second traitement de données à l'aide de l'installation de processeur qui établit initialement au moins une copie locale sous la forme d'une variable globale d'au moins une variable globale utilisée par la première installation et qui, lors de la copie, peut être interrompue par le premier traitement de données,
- on affiche l'opération de copie en cours du second traitement de données d'une manière lisible par le premier traitement de données et,
- on contrôle, dans le premier traitement de données, si l'affichage est mis à l'état, avant la fin d'une interruption respective du second traitement de données et, dans l'affirmative, le dépassement d'au moins une copie locale sous la forme de variables locales avec leur valeur actuelle.

7°) Procédé selon la revendication 6,

caractérisé par les étapes suivantes :

- on attribue une première priorité élevée à une première installation pour l'utilisation de l'installation de processeur et,
- 5 - on associe une seconde priorité faible à la seconde installation pour l'utilisation de l'installation de processeur.

8°) Procédé selon l'une quelconque des revendications 6 ou 7, caractérisé par les étapes suivantes :

- 10 - on associe une première durée courte à la première installation pour l'utilisation de l'installation de processeur et,
- on associe une seconde durée longue à la seconde installation pour l'utilisation de l'installation de processeur.

15

9°) Procédé selon l'une quelconque des revendications 6, 7, 8, caractérisé par l'étape suivante :

- on poursuit avec le second traitement de données, à la fin d'une interruption respective, à l'endroit de l'interruption.

20

10°) Procédé selon l'une quelconque des revendications 6 à 9, caractérisé par l'étape suivante :

- on détermine dans le premier traitement de données les copies locales utilisées par la seconde installation.

25

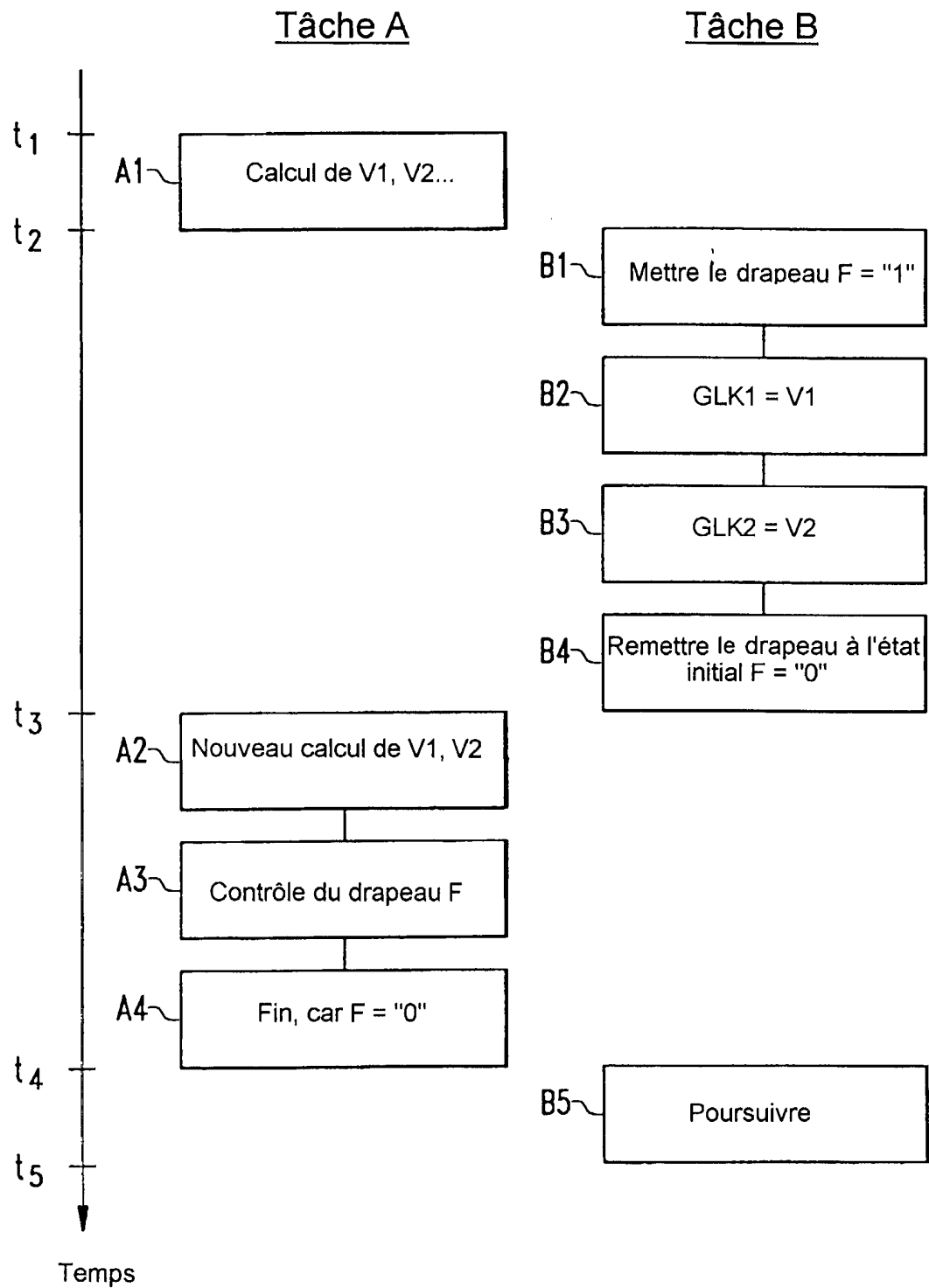


Fig.1

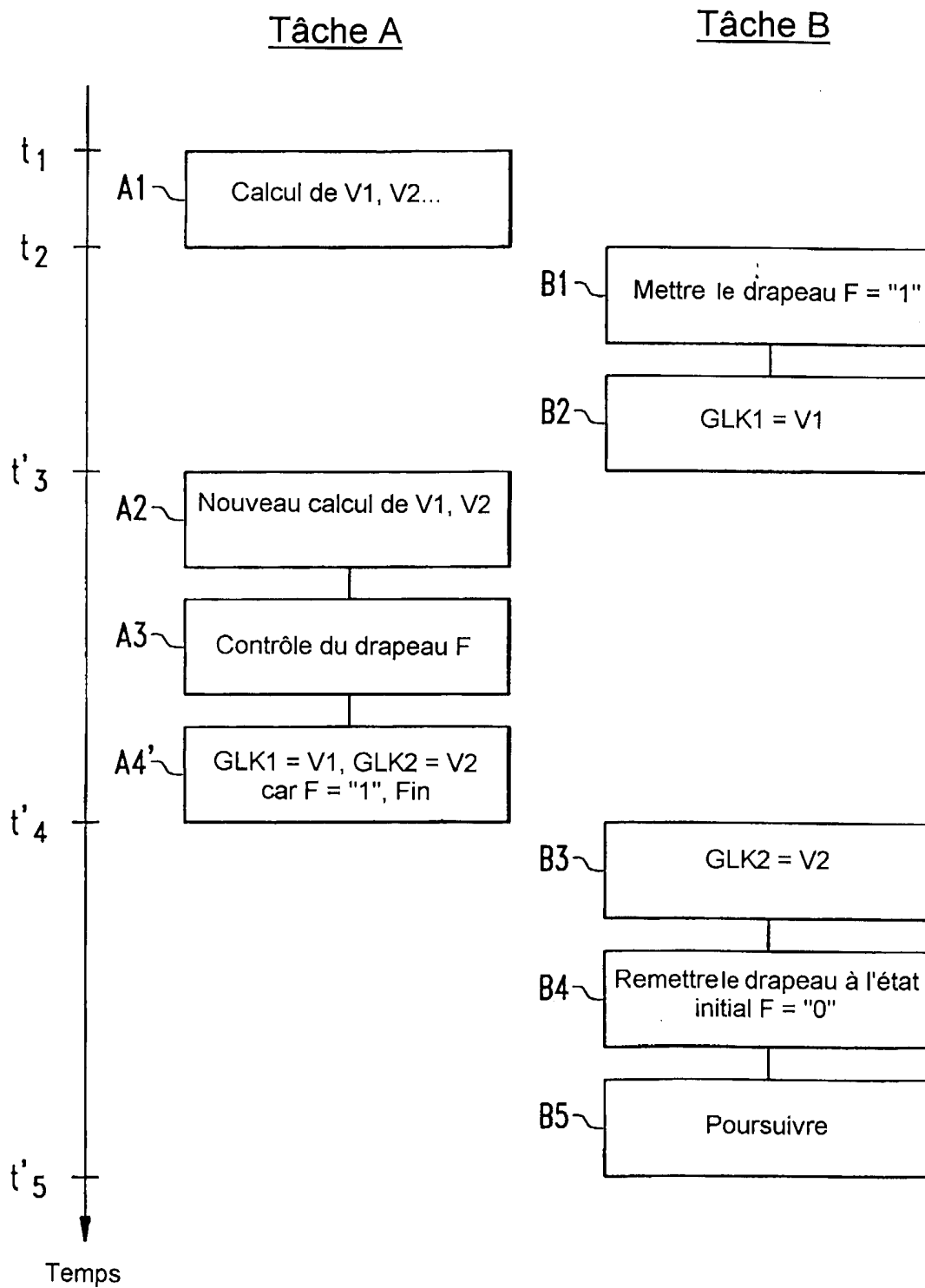


Fig.2

3/3

Tâche A (Tranche d'exécution 10 ms)

```

      |
S1:  V1 = - - - }
S2:  V2 = - - - } Calcul des données
      |
S4:  Si (Drapeau = 1) Alors }
S5:  GLK1 = V1                }
S6:  GLK2 = V2                } Répéter la copie
S7:  Fin Si                   }
      |
      Fin

```

Fig.3

Tâche B (Tranche d'exécution 50 ms)

```

      |
S1': Drapeau = 1 }
S2': GLK1 = V1   }
S3': GLK2 = V2   } Opération de copie
S4': Drapeau = 0 }
      |
S5': - - - = FONCTION (GLK1, GLK2) Poursuivre le traitement des données
      |
      FIN

```

Fig.4