



(19)
 Bundesrepublik Deutschland
 Deutsches Patent- und Markenamt

(10) DE 102 29 520 A1 2004.01.15

(12)

Offenlegungsschrift

(21) Aktenzeichen: 102 29 520.4
 (22) Anmeldetag: 01.07.2002
 (43) Offenlegungstag: 15.01.2004

(51) Int Cl.7: G06F 12/00
 B60R 16/02

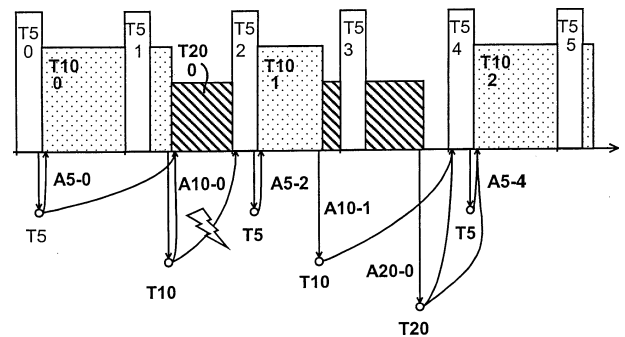
(71) Anmelder:
 Robert Bosch GmbH, 70469 Stuttgart, DE

(72) Erfinder:
 Hartmann, Uwe, 70839 Gerlingen, DE

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Verfahren und Vorrichtung sowie Betriebssystem zur Steuerung von Vorgängen bei einem Fahrzeug**

(57) Zusammenfassung: Verfahren und Vorrichtung sowie entsprechendes Betriebssystem zur Steuerung von Vorgängen bei einem Fahrzeug, wobei die Vorgänge durch wenigstens ein schnelleres Aufgabenprogramm und wenigstens ein langsamerer Aufgabenprogramm gesteuert werden, wobei wenigstens eine Ausgabegröße in dem wenigstens einen schnelleren Aufgabenprogramm und dem wenigstens einen langsameren Aufgabenprogramm verwendet wird und dazu von einem Speicherbereich in einen anderen Speicherbereich kopiert wird, wobei eine Ausgabegröße für das wenigstens eine langsamere Aufgabenprogramm am Ende des wenigstens einen schnelleren Aufgabenprogramms kopiert wird und/oder eine Ausgabegröße für das wenigstens eine schnellere Aufgabenprogramm zu Beginn des wenigstens eines schnelleren Aufgabenprogramms kopiert wird, wenn eine solche Ausgabegröße entsprechend für beide Aufgabenprogramme vorgesehen ist.



Beschreibung

Stand der Technik

[0001] Die Erfindung betrifft ein Verfahren und eine Vorrichtung sowie ein entsprechendes Betriebssystem zur Steuerung von Vorgängen bei einem Fahrzeug gemäß den Oberbegriffen der unabhängigen Ansprüche.

[0002] Bei Steuervorgängen im Fahrzeug liegen Betriebssysteme als Basis zur Ausführung von Steuervorgängen zugrunde. Bei Betriebssystemen, insbesondere Echtzeit-Betriebssystemen, wie sie z. B. in embedded-systems in der Automobilindustrie eingesetzt werden, sind die Funktionen oft auf mehrere Tasks sogenannte Aufgabenprogramme mit unterschiedlichen Abtastzeiten aufgeteilt, damit zeitkritische Teile öfter gerechnet werden können als weniger zeitkritische Algorithmen bzw. Programme oder Programmteile. Dabei sorgt im Normalfall das Betriebssystem für die Sicherstellung der Datenkonsistenz. So werden von Größen, auf die in mehreren Tasks zugegriffen wird, zu geeigneten Zeitpunkten Kopien angelegt, damit sich z. B. in langsamen Aufgabenprogrammen bzw. Tasks zwischen zwei Zugriffen der Wert nicht ändert, weil in einer schnelleren Task auf die Größe geschrieben wird. Dies ist beispielsweise in der EP 0 799 441 B1 gezeigt.

[0003] In regelungstechnischen Anwendungen, wie z. B. beim Fahrdynamikregelsystem ESP oder auch im Rahmen einer Motor- sowie Getriebesteuerung, usw. genügt es nicht, nur die Datenkonsistenz, wie oben beschrieben zu garantieren. Zusätzlich ist hier wichtig, dass die Kopien von Größen die in unterschiedlichen Tasks bzw. Aufgabenprogrammen verwendet werden zu bestimmten Zeitpunkten erstellt werden, um eine äquidistante Abtastung sicherzustellen; d. h. die Kopien sollten in immer möglichst gleichen zeitlichen Abständen erstellt werden. Zur Zeit verfügbare Echtzeit-Betriebssysteme für eingebettete Systeme sogenannte embedded-systems enthalten dafür keine Mechanismen.

[0004] Es stellt sich somit die Aufgabe, für regelungstechnische Anwendungen im Fahrzeug in einfacher Weise eine zeitlich äquidistante Abtastung zu ermöglichen. Als weitere Aufgabe sollen dann auch besonders zeitkritische Vorgänge ausreichend berücksichtigt werden können.

Vorteile der Erfindung

[0005] Durch die im folgenden vorgestellte Erfindung wird sichergestellt, dass in eingebetteten Systemen sogenannten embedded-systems mit Echtzeit-Betriebssystem alle Größen, die in mehreren Aufgabenprogrammen oder Tasks, mit unterschiedlichen Abtastzeiten benötigt werden, in zeitlich gleichen Abständen also äquidistant zwischen den Aufgabenprogrammen (Tasks) zur Sicherung der Datenkonsistenz kopiert werden, also ein ungewolltes

Überschreiben dieser Größen verhindert wird.

[0006] Für regelungstechnische Anwendungen ist die zeitlich äquidistante Abtastung oft unabdingbar, um Berechnungen wie beispielsweise Integration oder Differentiation von Signalen ohne erhöhten Rechenaufwand durchzuführen. Dieser erhöhte Rechenaufwand entstünde dabei beispielsweise durch die Einbeziehung von mitgelieferten Zeitinformatio- nen pro Signal.

[0007] Somit findet vorteilhafter Weise ein Verfahren und eine Vorrichtung sowie ein Betriebssystem zur Steuerung von Vorgängen bei einem Fahrzeug Einsatz, wobei die Vorgänge durch wenigstens ein schnelleres Aufgabenprogramm und wenigstens ein langsamerer Aufgabenprogramm gesteuert werden und wenigstens eine Ausgabegröße in dem wenigstens einen schnelleren Aufgabenprogramm und dem wenigstens einen langsameren Aufgabenprogramm verwendet wird, wobei diese dazu von einem Speicherbereich in einen anderen Speicherbereich kopiert wird und die eine Ausgabegröße für das wenigstens eine langsamere Aufgabenprogramm am Ende des wenigstens einen schnelleren Aufgabenprogramms kopiert wird und/oder eine Ausgabegröße für das wenigstens eine schnellere Aufgabenprogramm zur Beginn des wenigstens einen schnelleren Aufgabenprogramms kopiert wird, wenn eine solche Ausgabegröße entsprechend für beide Aufgabenprogramme vorgesehen ist.

[0008] Zweckmäßiger Weise ist dazu jedem Aufgabenprogramm ein eigener Speicherbereich und/oder jeweils ein interner Zähler zugeordnet.

[0009] Ebenso zweckmäßiger Weise erhalten die Aufgabenprogramme entsprechend einer maximalen Laufzeit jedes Aufgabenprogramms unterschiedliche Priorität.

[0010] Von Vorteil ist, dass nach Ablauf des langsamsten Aufgabenprogramms alle Zähler zu Beginn des schnellsten Aufgabenprogramms zurückgesetzt werden, und jeder Ablauf jedes Aufgabenprogramms im jeweiligen Zähler gezählt wird.

[0011] Zweckmäßiger Weise wird anhand einer Abfrage der Zählerstände zu Beginn und am Ende jedes Aufgabenprogramms wenigstens eine Kopieroutine aufgerufen, wobei vorteilhafter Weise anhand einer Vorschrift, insbesondere anhand einer Tabelle vorgegeben wird, welche Ausgabegröße wohin kopiert wird, also welche Kopieroutinen jeweils aufgerufen werden.

[0012] In einer vorteilhaften Ausgestaltung ist zusätzlich wenigstens ein Zustandsbit vorgesehen, welches durch das langsamere Aufgabenprogramm setzbar ist, wobei bei gesetztem Zustandsbit das schnellere Aufgabenprogramm direkt auf die Ausgabegröße also das Original der Ausgabegröße und nicht auf die Kopie des langsameren Aufgabenprogramms zugreift, eben bevor diese Kopie erstellt wird.

[0013] Zweckmäßiger Weise wird in dieser besonderen Ausgestaltung der Erfindung das Zustandsbit

dann zurückgesetzt, wenn die entsprechende Ausgabegröße des langsameren Aufgabenprogramms kopiert wird.

[0014] Weitere Vorteile und vorteilhafte Ausgestaltungen ergeben sich aus der Beschreibung sowie den Merkmalen der Ansprüche.

[0015] Zeichnung

[0016] Die Erfindung wird im Weiteren anhand der in der Zeichnung dargestellten Figuren näher erläutert.

[0017] Dazu zeigt **Fig. 1** eine schematische Darstellung einer Steuerung in einem Fahrzeug mit verschiedenen Aufgabenprogrammen (Tasks).

[0018] **Fig. 2** verdeutlicht das erfindungsgemäße Verfahren anhand dreier Aufgabenprogramme.

[0019] In **Fig. 3** ist eine Tabelle als Vorschrift zur Steuerung der Kopier- bzw. Übergaberoutinen angegeben.

[0020] **Fig. 4** zeigt eine besondere Ausgestaltung der Erfindung bei welcher ein Zustandsbit gesetzt und rückgesetzt wird.

Beschreibung der Ausführungsbeispiele

[0021] **Fig. 1** zeigt eine Steuerung **100** mit Eingängen bzw. Eingangsgrößen E und Ausgängen bzw. Ausgangsgrößen A. Dabei kann die Steuerung **100** als singuläres Steuergerät ebenso wie als Netzwerk busverbundener Steuereinheiten aufgebaut sein.

[0022] Die Steuerung kann weiterhin über Leitungen, insbesondere wenigstens ein Bussystem mit anderen Steuerungen sowie mit Sensorik oder Aktuatorik zur Steuerung von Vorgängen in einem Fahrzeug verbunden sein. Dabei von Sensorik, weiteren Steuerungen oder Aktuatorik gelieferte Eingangsgrößen sind in einem Block zusammengefasst und mit E bezeichnet.

[0023] Die durch die Steuerung **100** ausgegebenen Größen beispielsweise Rückantworten an andere Steuerungen oder Sensorik sowie Aktuatorik insbesondere Steuergrößen werden als Ausgangsgrößen A in einem Block zusammengefasst. In der Steuerung werden, insbesondere im Rahmen eines Betriebssystems, drei Aufgabenprogramme sogenannte Tasks T5, T10 und T20 betrachtet. Diese können abhängig von der Struktur der Steuerung **100** in einem Steuergerät oder in mehreren Steuereinheiten vorhanden sein. Dabei ist es sogar denkbar, eine einzelne Task über mehrere Steuereinheiten zu verteilen. Gleiches gilt für die den einzelnen Aufgabenprogrammen oder Tasks zugeordneten Speicherbereiche SB5, SB10 und SB20. Diese sind jeweils einer Task zugeordnet, können aber sowohl im gleichen Steuergerät bzw. im gleichen Speicher oder auch in verschiedenen Speichern und/oder verschiedenen Steuergeräten, sozusagen als virtueller Speicher, verteilt sein.

[0024] Den einzelnen Aufgabenprogrammen oder Tasks T5, T10 und T20 sind jeweils Zähler Z5, Z10 und Z20 zugeordnet, die einerseits in Hardware oder

Software realisiert sein können und als interner Zähler der entsprechenden Task fungieren. Die einzelnen Aufgabenprogramme oder Tasks können dabei auch unter einander Variablen bzw. Größen insbesondere im Rahmen der einzelnen Programmaufgaben austauschen. Da diese Variablen oder Größen im allgemeinen ihre Ausgangspunkte in einer bestimmten Task haben und von dieser ausgegeben werden, entweder als Ausgangsgröße A oder an andere Tasks werden diese Größen im weiteren allgemein als Ausgabegrößen bezeichnet.

[0025] In **Fig. 2** wird als Beispiel die Vorgehensweise bei drei Aufgabenprogrammen oder drei Tasks T5, T10 und T20 gezeigt. Zur Sicherung der Datenkonsistenz ist es notwendig, dass die Größen, nachfolgend als Ausgabegrößen bezeichnet, die in mehreren Tasks bzw. Aufgabenprogrammen verwendet werden, im Normalfall auch mehrfach im Speicher beispielsweise einem RAM-Speicher abgelegt werden. D. h. der Speicher ist hier beispielsweise als RAM ausgeführt, jegliche weitere dem Fachmann bekannte Speicherform wie oben erwähnt auch virtuell verteilt in einem Netzwerk kann dabei verwendet werden.

[0026] Werden diese Größen, diese Ausgabegrößen nicht mehrfach im Speicher abgelegt, dann kann es z. B. vorkommen, dass sich Größen die in einer schnellen Task geschrieben werden und in einer langsameren Task gelesen werden, während der Laufzeit der langsameren Task ändern. Das bedeutet, dass jedes Aufgabenprogramm, also jede Task vorteilhafter Weise ihren eigenen Speicherbereich SB besitzt. Zu geeigneten Zeitpunkten müssen dabei die Variablen, also hier die Ausgabegrößen umkopiert werden, wenn diese Ausgabegrößen für mehrere Aufgabenprogramme vorgesehen sind. Dazu könnte z. B. eine Größe, die in der Task T10 beispielsweise einer 10 ms-Task, also mit einer maximalen Laufzeit von 10 ms, berechnet wird, an die Task T5, also beispielsweise einer Task mit einer maximalen Laufzeit von 5 ms übergeben werden und zwar an diejenige Task T5, die nach Beendigung der entsprechenden 10 ms-Task T10 als nächstes startet. Dies führt dann jedoch zu nicht äquidistanten Abtastungen, da im Beispiel die Task T10, also die 10 ms-Task unterschiedlich lange Laufzeiten haben kann. Deshalb wird die Kopie der Variablen zu festgelegten Zeitpunkten erzeugt, mit dem Ziel, generell äquidistante Abtastungen sicherzustellen.

[0027] Dazu startet in **Fig. 2** zunächst die T5-Task, also hier im Beispiel 5 ms, da sie in diesem Beispiel die höhere Priorität gegenüber langsameren Tasks, also hier T10 und T20 besitzt. D. h. zweckmäßiger Weise ist hier eine Priorität abhängig von der maximalen Laufzeit bzw. der Schnelligkeit des Aufgabenprogramms eingeführt. Am Ende der T5-Task werden die Schnittstellengrößen, d. h. in diesem Fall die in der T5-Task T5-0 berechneten Ausgabegrößen A5-0 auf den Speicherbereich SB10 der Task T10 bzw. SB20 der Task T20 umkopiert.

[0028] Nach Beendigung der Task T5-0 startet die 10 ms-Task T10-0. Diese Task T10-0 wird durch die nächste 5 ms-Task T5-1 auf grund der höheren Priorität unterbrochen. Am Ende der Task T10-0 werden die Ausgabegrößen A10-0 der Task T10-0 auf die jeweiligen Speicherbereiche der Task T20, also nach SB20 umkopiert, nicht jedoch auf den Speicherbereich der schnelleren T5-Task T5-2. Würde dieses Umkopieren der Ausgabegrößen A10-0 auf SB5, also für T5-2 durchgeführt, dann erhielte im Beispiel die Task T5-2 die Ausgabegröße A10-0 der Task T10-0 und bereits die nächste 5 ms-Task T5-3 die Ergebnisse von T10-1, weil die zweite 10 ms-Task zufällig etwas früher beendet wird als die erste. Das Prinzip der äquidistanten Abtastung wäre in diesem Fall verletzt. Aus diesem Grund wird die Kopieroutine von einer langsamen zu einer schnelleren Task immer zur Beginn der schnelleren Task durchgeführt, d. h. zur Beginn der Task T5-2 werden die Ausgabegrößen A10-0 von T10-0 kopiert und zur Beginn der Task T5-4 diejenigen von T10-1 und T20-0.

[0029] D. h. zur Sicherstellung der äquidistanten Abtastung lauten die Kopierregeln also, dass die Ausgabegrößen für langsamere Tasks am Ende der schnelleren Task umkopiert werden und Ausgabegrößen für schnellere Tasks zur Beginn der schnelleren Task umkopiert werden.

[0030] Zur Steuerung der Kopieroutinen, also der Vorschrift welche Ausgabegröße bzw. welche Ausgabegrößen von welcher Task zu welcher anderen Task kopiert werden, besitzen die Tasks, also hier T5, T10 und T20 jeweils einen internen Zähler bzw. ist diesen ein Zähler zugeordnet, gemäß Z5, Z10 und Z20 in **Fig. 1**. Hier im Beispiel werden nach 20 ms alle Zähler zu Beginn einer T5-Task auf Null gesetzt, bzw. auf den Anfangswert zurückgesetzt. Am Ende jeder Task wird der entsprechende Zähler um Eins erhöht. Durch Abfrage der Zählerstände zu Beginn und am Ende jeder Task kann dann gesteuert werden, welche Kopieroutinen jeweils aufgerufen werden.

[0031] Dies ist beispielsweise anhand von Tabellen gemäß **Fig. 3** darstellbar. Hier sind für den T5-Zähler Z5 und den T10-Zähler Z10 in Tabellen 300 und 301 die Kopiervorschriften abhängig vom Zählerstand dargestellt. So werden bei T5-Zählerstand, also Z5-Zählerstand 0 und zu Beginn der Task T5 die Ausgabegrößen von T10 auf T5 kopiert sowie von T20 auf T5 und am Ende der T5-Task die Ausgabegrößen von T5 auf T10 und T5 auf T20. Die übrigen Umkopiervorgänge sind ebenfalls beispielhaft wie gerade beschrieben, den Tabellen **300** und **301** zu entnehmen.

[0032] So werden am Ende von T10-1 die Ausgabegrößen A10-1 wie beschrieben auf T5-4 übertragen. Ebenso werden die Größen A20-0 am Ende der Task T20-0 auf T5-4 sowie T10-2 übertragen. Schließlich wird noch A5-4 von T5-4 auf T10-2 umkopiert, wobei dies anhand der in **Fig. 3** dargestellten Vorschriften gesteuert wird.

[0033] Somit ist für regelungstechnische Anwen-

dungen eine zeitlich äquidistante Abtastung ohne Einbeziehung von mitgelieferte Zeitinformation, also ohne erhöhten Rechenaufwand sehr einfach durchzuführen.

[0034] In manchen zeitkritischen Fällen dargestellt in **Fig. 4**, kann in einer schnelleren Task nicht gewartet werden, bis die Kopie einer Ausgabegröße aus einer langsameren Task vorliegt, sondern man möchte auf das Ergebnis einer Berechnung zugreifen, sobald dieses vorliegt, also auf das Original der Ausgabegröße, u.U. unter Verzicht auf die äquidistante Abtastung.

[0035] In dieser speziellen Ausgestaltung der Erfindung eben ein kurzfristiges Aufheben der äquidistanten Abtastung ohne insgesamt die äquidistante Abtastung zu verletzen oder in Frage zu stellen setzt die langsamere Task ein Zustandsbit, sobald eine oder mehrere Größen vollständig berechnet wurden. Bei gesetztem Zustandsbit darf die schnellere Task direkt auf das Original der Variablen bzw. der Ausgabegröße zugreifen.

[0036] Beim nachfolgenden standardmäßigen Kopiervorgang wird das Zustandsbit dann wieder zurückgesetzt. Nun, also bei zurückgesetztem Zustandsbit darf nur die eben erzeugte Kopie der Ausgabegröße in der schnelleren Task verwendet werden. Zwischen dem Setzen des Zustandsbits und der Ausführung der Kopieroutine darf die Ausgabegröße dann nicht mehr verändert werden.

[0037] Bei gesetztem Zustandsbit ist also jeweils das Original gültig, weil es in der langsameren Task fertig berechnet wurde und vor Ablauf der Task nicht mehr geändert wird. Bei nicht gesetztem Zustandsbit ist nur die Kopie gültig.

[0038] Diese zeitliche Abfolge ist eben in **Fig. 4** dargestellt. Im Beispiel muss die Task T5-1 mit der Kopie der Ausgabegröße arbeiten, weil die Originalgröße in der 10 ms-Task T10-0 noch nicht fertig berechnet wurde und zur Zeit möglicherweise ungültige Werte aufweisen kann. Erst wenn das Ergebnis vorliegt, also das Original der Ausgabegröße AO10-0 fertig berechnet wurde, wird das Zustandsbit gesetzt und es kann auf das Original AO10-0 zugegriffen werden. Anschließend wird die Kopie der Ausgabegröße A10-0 von T10-0 zur Task T5-2 umkopiert.

[0039] Bei diesem Kopiervorgang wird nun das Zustandsbit wieder zurückgesetzt. Gleiches gilt dann für die nächste Größe AO10-1, also dem Original der Task T10-1 zu dem Zeitpunkt in dem dieses fertig berechnet bzw. bereitgestellt wurde. Auch hier kann auf dieses Original zugegriffen werden und das Zustandsbit wird wiederum gesetzt. Erst bei nachfolgender Erstellung der Kopie A10-1 für T5-4 wird dann wiederum das Zustandsbit rückgesetzt. Gleiches gilt dann auch für AO10-2.

[0040] Die Tasks T5-2 und T5-4 verwenden die bzw. rechnen mit der gerade erstellten Kopie A10-0 bzw. A10-1, wie in der Erfindung vorgesehen und nicht mit dem Original. Die Task T5-3 hat über diese zusätzliche Option des Zustandsbits aber die Möglichkeit be-

reits mit der fertig berechneten Originalgröße AO10-1 aus der 10 ms-Task T10-1 weiter zu rechnen. Ebenso gilt dies für die Task T5-5 mit der Originalgröße AO10-2 der Task T10-2.

[0041] Somit ermöglicht die Erfindung auf einfache Weise eine äquidistante Abtastung, wobei in einer speziellen Ausgestaltung zusätzlich in besonders zeitkritischen Situationen durch einen Zusatzmechanismus ohne Verlust der Sicherheit der Mechanismus der äquidistanten Abtastung kurzfristig umgangen werden kann, ohne diesen grundsätzlichen, erfindungsgemäßen Mechanismus der äquidistanten Abtastung mit Fehlern zu belegen. D. h. es wird im Normalfall die äquidistante Abtastung garantiert, wobei trotzdem zeitkritische Ausnahmen mit hoher Sicherheit berücksichtigt werden können.

Patentansprüche

1. Verfahren zur Steuerung von Vorgängen in einem Fahrzeug, wobei die Vorgänge durch wenigstens ein schnelleres Aufgabenprogramm und wenigstens ein langsamerer Aufgabenprogramm gesteuert werden, wobei wenigstens eine Ausgabegröße in dem wenigstens einen schnelleren Aufgabenprogramm und dem wenigstens einen langsameren Aufgabenprogramm verwendet wird und dazu von einem Speicherbereich in einen anderen Speicherbereich kopiert wird, **dadurch gekennzeichnet**, dass eine Ausgabegröße für das wenigstens eine langsamere Aufgabenprogramm am Ende des wenigstens einen schnelleren Aufgabenprogramms kopiert wird und/oder eine Ausgabegröße für das wenigstens eine schnellere Aufgabenprogramm zu Beginn des wenigstens einen schnelleren Aufgabenprogramms kopiert wird, wenn eine solche Ausgabegröße für beide Aufgabenprogramme vorgesehen ist.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass jedem Aufgabenprogramm ein eigener Speicherbereich zugeordnet ist.

3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass jedem Aufgabenprogramm jeweils ein interner Zähler zugeordnet ist.

4. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Aufgabenprogramme unterschiedliche Priorität abhängig von einer maximalen Laufzeit jedes Aufgabenprogramms besitzen.

5. Verfahren nach Anspruch 3, dadurch gekennzeichnet, dass nach Ablauf des langsamsten Aufgabenprogramms alle Zähler zu Beginn des schnellsten Aufgabenprogramms zurückgesetzt werden und jeder Ablauf jedes Aufgabenprogramms im jeweiligen Zähler gezählt wird.

6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, dass anhand einer Abfrage der Zählerstän-

de zu Beginn und am Ende jedes Aufgabenprogramms wenigstens eine Kopieroutine aufgerufen wird.

7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, dass anhand einer Vorschrift, insbesondere anhand einer Tabelle, vorgegeben wird welche Ausgabegröße wohin kopiert wird, also welche Kopieroutinen jeweils aufgerufen werden.

8. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass wenigstens ein Zustandsbit durch das langsamere Aufgabenprogramm setzbar ist und dass bei gesetztem Zustandsbit das schnellere Aufgabenprogramm direkt auf die Ausgabegröße des langsameren Aufgabenprogramms zugreift bevor diese kopiert wird.

9. Verfahren nach Anspruch 8, dadurch gekennzeichnet, dass dann, wenn die Ausgabegröße des langsameren Aufgabenprogramms kopiert wird, das Zustandsbit zurückgesetzt wird.

10. Betriebssystem zur Steuerung von Vorgängen in einem Fahrzeug, bei welchem durch Ausführung von Aufgabenprogrammen wenigstens einer der Verfahrensschritte nach einem der Ansprüche 1 bis 9 ausgeführt wird.

11. Vorrichtung mit wenigstens einem Speicher zur Steuerung von Vorgängen in einem Fahrzeug, wobei erste Mittel enthalten sind, welche die Vorgänge durch wenigstens ein schnelleres Aufgabenprogramm und wenigstens ein langsamerer Aufgabenprogramm steuern, wobei wenigstens eine Ausgabegröße in dem wenigstens einen schnelleren Aufgabenprogramm und dem wenigstens einen langsameren Aufgabenprogramm verwendet wird und dazu von einem Speicherbereich des Speichers in einen anderen Speicherbereich kopiert wird, dadurch gekennzeichnet, dass zweite Mittel enthalten sind, die eine Ausgabegröße für das wenigstens eine langsamere Aufgabenprogramm am Ende des wenigstens einen schnelleren Aufgabenprogramms kopieren und/oder eine Ausgabegröße für das wenigstens eine schnellere Aufgabenprogramm zu Beginn des wenigstens einen schnelleren Aufgabenprogramms kopieren, wenn eine solche Ausgabegröße für beide Aufgabenprogramme vorgesehen ist.

Es folgen 4 Blatt Zeichnungen

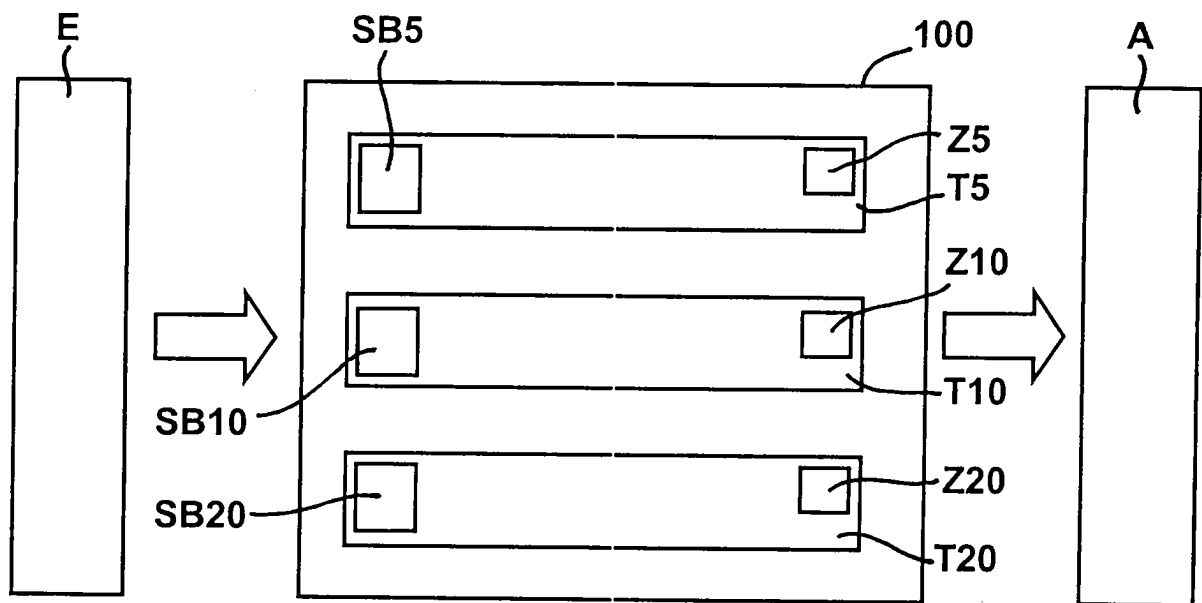


Fig. 1

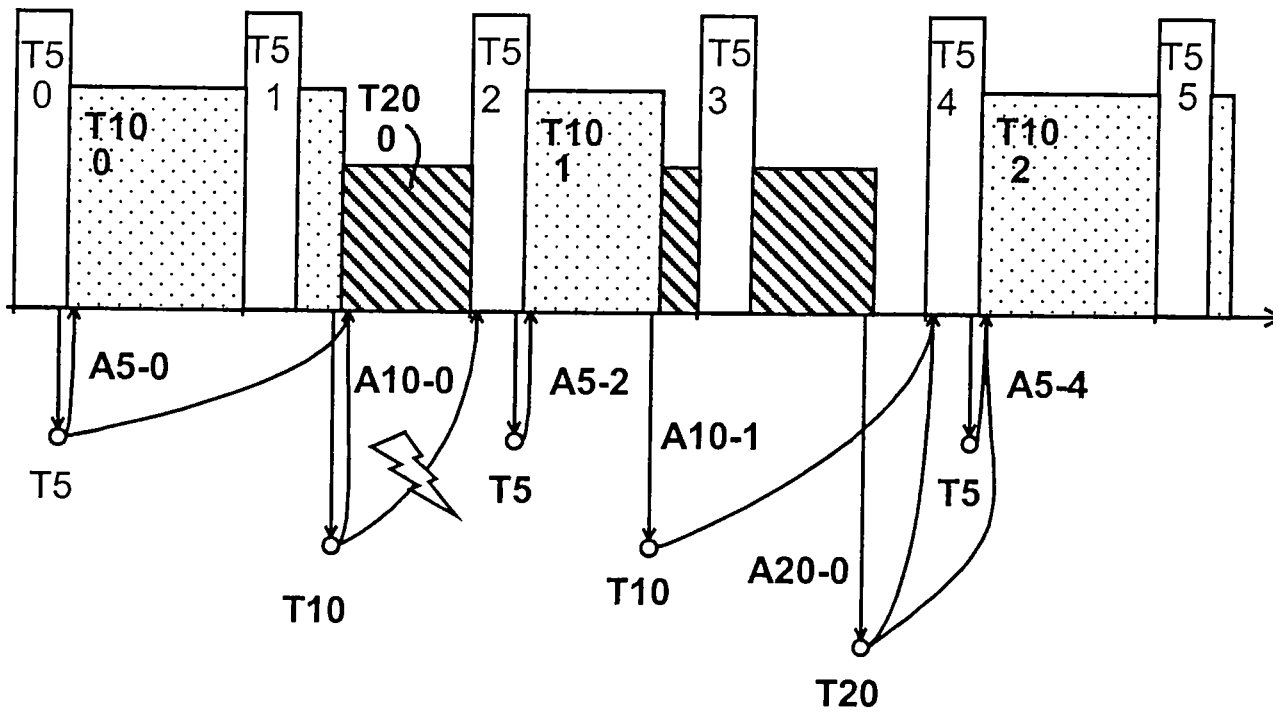


Fig. 2

300

T5-Zähler	Zu Beginn der T5-Task		Am Ende der T5-Task	
	0	T10 → T5	T20 → T5	T5 → T10
1	-	-	-	-
2	T10 → T5	-	T5 → T10	-
3	-	-	-	-

301

T10-Zähler	Zu Beginn der T10-Task	Am Ende der T10-Task
	0	T20 → T10
1	-	-

Fig. 3

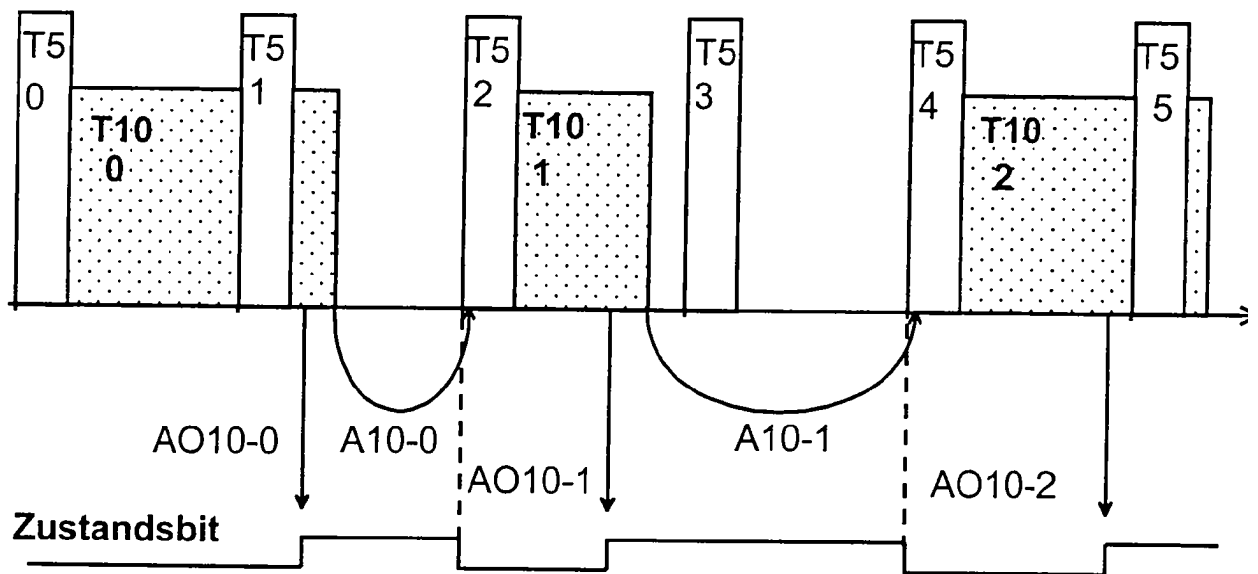


Fig. 4