

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES
PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro

(43) Internationales Veröffentlichungsdatum
06. August 2020 (06.08.2020)



(10) Internationale Veröffentlichungsnummer
WO 2020/156668 A1

(51) Internationale Patentklassifikation:
G06F 8/30 (2018.01)

(21) Internationales Aktenzeichen: PCT/EP2019/052315

(22) Internationales Anmeldedatum:
31. Januar 2019 (31.01.2019)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(71) Anmelder: **BAYERISCHE MOTOREN WERKE AK-
TIENGESSELLSCHAFT** [DE/DE]; Petuelring 130, 80809
München (DE).

(72) Erfinder: **REUTER, Jakob**; Simmerleinplatz 8, 80992
München (DE). **BORCHERS, Christoph**; Memeler Str.
90, 81929 München (DE).

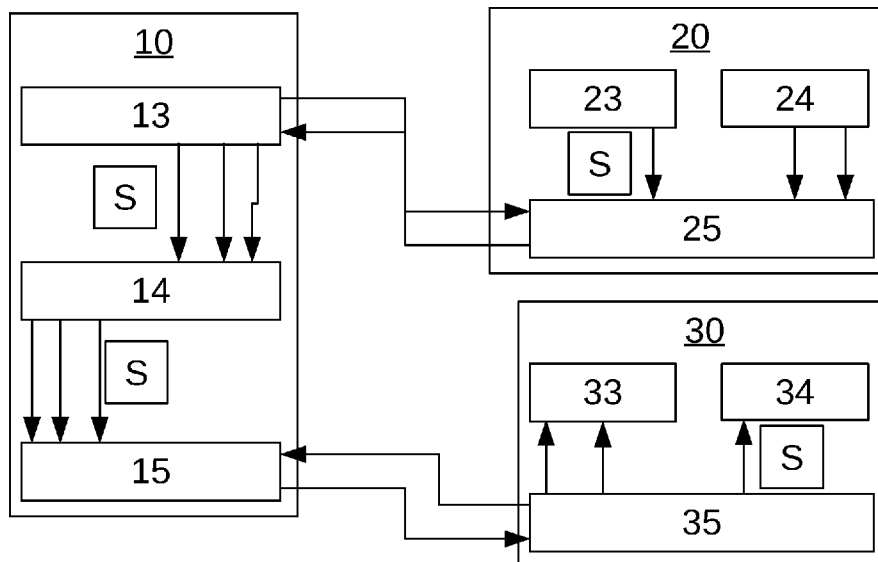
(81) Bestimmungsstaaten (soweit nicht anders angegeben, für
jede verfügbare nationale Schutzrechtsart): AE, AG, AL,
AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,
BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN,
KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD,
ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO,
NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW,
SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM,
TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Bestimmungsstaaten (soweit nicht anders angegeben, für
jede verfügbare regionale Schutzrechtsart): ARIPO (BW,
GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST,
SZ, TZ, UG, ZM, ZW), eurasisches (AM, AZ, BY, KG, KZ,
RU, TJ, TM), europäisches (AL, AT, BE, BG, CH, CY, CZ,
DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT,

(54) Title: METHOD FOR EXECUTING PROGRAM COMPONENTS ON A CONTROL UNIT, A COMPUTER-READABLE
STORAGE MEDIUM, A CONTROL UNIT AND A SYSTEM

(54) Bezeichnung: VERFAHREN ZUM AUSFÜHREN VON PROGRAMMKOMPONENTEN AUF EINEM STEUERGERÄT, EIN
COMPUTERLESBARES-SPEICHERMEDIUM, EIN STEUERGERÄT UND EIN SYSTEM

Fig. 3



(57) Abstract: The invention relates to a method for executing program components (23, 24, 33, 34) on a control unit (2), comprising the following steps: - receiving at least one first program unit (8) and at least one second program unit (9, 9'); - producing a first and a second proxy definition, wherein a proxy definition stipulates access to at least one function and/or a memory area of a program unit (8, 9, 9'), wherein the first proxy definition is associated with the first program unit (8) and the second proxy definition (9) is associated with the second program unit (9, 9'); - compiling the first program unit (8) and the second program unit (9, 9') to produce a first program component (23), a second program component (24) and a first and a second proxy component (13, 25); - executing the first program

LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI,
SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN,
GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Veröffentlicht:

- mit internationalem Recherchenbericht (Artikel 21 Absatz
3)

component (23) and the second program component (24) on a control unit (2), wherein the first program component (23) calls and/or uses at least one function of the second program component (24) by using the first and the second proxy component (13, 25).

(57) Zusammenfassung: Die Erfindung betrifft ein Verfahren zum Ausführen von Programmkomponenten (23, 24, 33, 34) auf einem Steuergerät (2), umfassend die folgenden Schritte: - Empfangen mindestens einer ersten Programmeinheit (8) und mindestens einer zweiten Programmeinheit (9, 9'); - Erzeugen einer ersten und einer zweiten Proxydefinition, wobei eine Proxydefinition einen Zugriff auf mindestens einer Funktion und/oder einen Speicherbereich einer Programmeinheit (8, 9, 9') festlegt, wobei die erste Proxydefinition der ersten Programmeinheit (8) und die zweite Proxydefinition (9) der zweiten Programmeinheit (9, 9') zugeordnet ist; - Kompilieren der ersten Programmeinheit (8) und der zweiten Programmeinheit (9, 9') zur Erzeugung einer ersten Programmkomponente (23), einer zweiten Programmkomponente (24) und einer ersten und einer zweiten Proxykomponente (13, 25); - Ausführen der ersten Programmkomponente (23) und der zweiten Programmkomponente (24) auf einem Steuergerät (2), wobei die erste Programmkomponente (23) mindestens eine Funktion der zweiten Programmkomponente (24) unter Verwendung der ersten und der zweiten Proxykomponente (13, 25) aufruft und/oder verwendet.

Verfahren zum Ausführen von Programmkomponenten auf einem Steuergerät, ein computerlesbares-Speichermedium, ein Steuergerät und ein System

Beschreibung

Die Erfindung betrifft ein Verfahren zum Ausführen von Programmkomponenten auf einem Steuergerät, ein computerlesbares-Speichermedium, ein Steuergerät und ein System.

Fahrzeuge, wie PKW, umfassen zur Steuerung der Fahrzeugfunktionen Steuergeräte. Traditionell wird dabei für jede Steuereinheit ein eigenes Steuergerät eingesetzt. Für die Querdynamik des Fahrzeugs oder für einzelne Fahrerassistenzsysteme wird also jeweils ein separates Steuergerät eingesetzt.

Ein Nachteil dabei ist, dass in dem begrenzten Bauraum des Fahrzeugs eine Vielzahl von Steuergeräten untergebracht werden muss. Dies ist nachteilig, da so weniger Flexibilität bei der Gestaltung der Fahrzeuge besteht. Ferner ist ein wesentlicher Nachteil, dass ein Steuergerät jeweils genau für die auszuführenden Funktionen ausgelegt ist, d.h. hinsichtlich des bereitgestellten Speichers und der Rechenkapazität. Dies ist nachteilig, da somit Funktionen nur im Rahmen der bestehenden Hardwaregrenzen per Softwareupdate nachgerüstet werden können.

In modernen Fahrzeugen werden daher zunehmend Universalsteuergeräte eingesetzt, die die Steuerung von mehreren Funktionen des Fahrzeugs übernehmen. Dazu weisen die Universalsteuergeräte mehrere Rechenkerne auf, wobei einer oder mehrere der Rechenkerne jeweils unterschiedlichen Funktionen zugewiesen werden können. Auch können Speicherbereiche unterschiedlichen Funktionseinheiten zugewiesen werden.

Dadurch, dass mehrere Funktionen auf einem (Universal)Steuergerät gebündelt werden, ist häufiger notwendig, ein Steuergerät zu aktualisieren, d.h. die Software zu aktualisieren. Die Universalsteuergeräte weisen unterschiedliche Partitionen auf, wobei auf eine der Partitionen ein Betriebssystem, wie z.B. ein Autosar OS-basiertes System, gespeichert ist, welches die Ausführung der weiteren Applikationen steuert. Auf einzelnen anderen Partitionen ist dann die jeweilige Applikation für eine Steuerfunktion des Steuergerätes gespeichert.

Beispielsweise kann auf einer ersten Partition ein Autosar OS Betriebssystem gespeichert sein. Auf einer zweiten Partition kann die Software zur Steuerung der Querdynamik und auf einer dritten Partition kann die Software zur Steuerung einer Fahrerassistenzfunktion gespeichert sein.

Bevor eine Funktion aktualisiert wird, muss geprüft werden, ob die zu aktualisierende Funktion sicherheitskritisch ist. Ist dies der Fall, so muss nach dem Aktualisieren eine Testfahrt mit einem Testfahrzeug ausgeführt werden. Dies nimmt viel Zeit in Anspruch und ist teuer.

Dieser Nachteil wird dadurch verstärkt, dass die bei Universalsteuergeräten eingesetzten Programmiersprachen üblicherweise erfordern, sämtliche Komponenten neu zu kompilieren, wenn sich auch nur eine Komponente ändert. Das bedeutet, dass häufig Programmabschnitte neu kompiliert werden, die nicht geändert sind. Dies führt zu einer Reihe von Testfahrten und Wartezeiten, die nicht notwendigerweise erforderlich wären.

Ausgehend von diesem Stand der Technik ist es daher Aufgabe der Erfindung, die Anzahl von notwendigen Testfahrten nach der Aktualisierung eines Steuergeräts zu reduzieren. Es ist insbesondere weitere Aufgabe der Erfindung, das Aktualisieren von Steuergeräten zu beschleunigen. Es ist weiter insbesondere Aufgabe der Erfindung die beim Kompilieren benötigte Rechenkapazität zu reduzieren.

Diese Aufgabe wird durch ein Verfahren nach Anspruch 1, ein computerlesbares Speichermedium nach Anspruch 10, ein Steuergerät nach Anspruch 11 und durch ein System nach Anspruch 12 gelöst

Insbesondere wird die Aufgabe durch ein Verfahren zum Ausführen von Programmkomponenten auf einem Steuergerät gelöst, insbesondere auf einem Steuergerät eines Fahrzeugs, bevorzugt eines PKWs, umfassend die folgenden Schritte:

- Empfangen mindestens einer ersten Programmeinheit und mindestens einer zweiten Programmeinheit;

- Erzeugen einer ersten und einer zweiten Proxydefinition, wobei eine Proxydefinition einen Zugriff auf mindestens eine Funktion und/oder einen Speicherbereich einer Programmeinheit festlegt, wobei die erste Proxydefinition der ersten Programmeinheit und die zweite Proxydefinition der zweiten Programmeinheit zugeordnet ist;
- Kompilieren der ersten Programmeinheit und der zweiten Programmeinheit zur Erzeugung einer ersten Programmkomponente, einer zweiten Programmkomponente und Kompilieren der ersten und zweiten Proxydefinition zur Erzeugung einer ersten und einer zweiten Proxykomponente;
- Ausführen der ersten Programmkomponente und der zweiten Programmkomponente auf einem Steuergerät, wobei die erste Programmkomponente mindestens eine Funktion der zweiten Programmkomponente unter Verwendung der ersten und der zweiten Proxykomponente aufruft und/oder verwendet.

Ein Kern der Erfindung ist, dass für die Programmeinheiten Proxydefinitionen erzeugt werden, die wiederum zu Proxykomponenten kompiliert werden. Unter Verwendung dieser Proxykomponenten können die erste und die zweite Programmkomponente jeweils Funktionen der anderen Programmkomponente aufrufen. Die Schnittstelle zwischen der ersten und der zweiten Programmkomponente wird also durch die erste und die zweite Proxykomponente gebildet. Bei einer Änderung nur einer Programmkomponente, bei der sich die Schnittstelle nicht ändert, kann daher auf ein Kompilieren der nicht geänderten Komponente verzichtet werden.

In einer Ausführungsform können die erste und die zweite Programmkomponente als Binärdateien ausgebildet sein, die jeweils durch einen Rechenkern eines Steuergeräts ausführbar sein können.

Die erste und die zweite Programmkomponente können demnach einzelne Binärdateien darstellen, die z.B. als Images auf Partitionen des Steuergeräts gespeichert sein können und somit durch einen Rechenkern des einen Steuergeräts ausführbar sind. Ein Rechenkern eines Steuergeräts kann im

Rahmen dieser Anmeldung auch ein Prozessorkern eines Prozessors eines Steuergeräts umfassen.

In einer Ausführungsform kann das Ausführen ein Ausführen der ersten Programmkomponente und/oder der ersten Proxykomponente auf einem ersten Rechenkern des Steuergeräts und ein Ausführen der zweiten Programmkomponente und/oder der zweiten Proxykomponente auf einem zweiten Rechenkern des Steuergeräts umfassen.

Es ist also möglich, dass unterschiedliche Rechenkerne unterschiedliche Programmkomponenten ausführen. Dabei kann eine Zuordnung von einer Programmkomponente zu einem Rechenkern dynamisch oder statisch festgelegt sein.

In einer Ausführungsform können die erste und/oder die zweite Programmeinheit Quellcode einer Programmiersprache, insbesondere embedded C, umfassen.

Embedded-C hat sich bei der hardwarenahen Programmierung von Steuergeräten bewährt. Somit wird eine leicht umzusetzende Implementierung angegeben.

In einer Ausführungsform können die erste und/oder die zweite Proxydefinition Adressen von Funktionen der ersten bzw. zweiten Programmkomponente angeben.

Besonders vorteilhaft zur Implementierung der Erfindung ist, wenn die Proxydefinitionen Angaben darüber enthalten, über welche Speicheradressen Funktionen von Programmkomponenten aufrufbar sind. Ebenso vorteilhaft ist es, wenn die Proxydefinitionen Angaben über gemeinsam genutzte Speicherbereiche enthalten. Mit der beschriebenen Ausführungsform wird eine einfache Art angegeben, die Schnittstelle zwischen der ersten und der zweiten Programmkomponente festzulegen.

In einer Ausführungsform kann das Verfahren die folgenden Schritte umfassen:

- Empfangen einer dritten Programmeinheit, die hinsichtlich ihrer Schnittstellen der zweiten Programmeinheit entspricht;

- Kompilieren nur der dritten Programmeinheit zur Erzeugung einer dritten Programmkomponente;
- Ausführen der ersten Programmkomponente und der dritten Programmkomponente auf einem Steuergerät, wobei die dritte Programmkomponente mindestens eine Funktion der ersten Programmkomponente unter Verwendung der ersten und der zweiten Proxykomponente aufruft und/oder verwendet.

Bei der dritten Programmeinheit kann es sich z.B. um eine aktualisierte Version der zweiten Programmeinheit handeln, bei der sich die Schnittstellen nicht geändert haben. Es ist nun nicht notwendig, die erste Programmeinheit neu zu kompilieren. Die dritte Programmkomponente kann unter Verwendung der ersten und zweiten Proxykomponenten auf die Funktionen und Speicherbereiche der ersten Programmkomponente zugreifen., sodass auch diese nicht erneut erzeugt werden müssen.

In einer Ausführungsform kann das Kompilieren der dritten Programmeinheit unter Verwendung eines Regelsatzes ausgeführt werden, wobei der Regelsatz mindestens eine Regel umfassen kann, die angeben kann, ob für ein Kompilieren der dritten Programmeinheit ein Kompilieren der ersten Programmeinheit notwendig ist.

Eine Regel kann Bedingungen angeben, wann einzelne Programmkomponenten bei Änderungen erneut kompiliert werden. So kann eine Regel in einer Ausführungsform angeben, dass eine Programmkomponente neu kompiliert werden muss, wenn sie eine Funktionsdefinition einer Funktion der Programmkomponente geändert hat. Eine weitere mögliche Regel kann in einer Ausführungsform angeben, dass eine Programmkomponente neu kompiliert werden muss, wenn sich die Größe eines gemeinsam genutzten Speicherbereichs (Shared Memory) geändert hat.

In einer Ausführungsform kann die erste Programmkomponente ein Betriebssystem für das Steuergerät umfassen und/oder die erste Komponente kann dazu ausgebildet sein, die Querdynamik eines Fahrzeugs zu steuern und/oder die dritte Programmkomponente kann dazu ausgebildet sein, mindestens ein Fahrerassistenzsystem eines Fahrzeugs zu steuern.

In einer vorteilhaften Ausführungsform kann durch die Proxykomponenten die Kommunikation zwischen der Steuerung der Querdynamik und der Steuerung von mindestens einem Fahrerassistenzsystem ausgeführt werden.

Bei diesen häufig aktualisierten Komponenten kommen die vorstehend beschriebenen Vorteile besonders zum Tragen, da eine häufige Aktualisierung sonst zu einer Kompilierung sämtlicher Komponenten führen würde, was viel Zeit kostet und weitere Testfahrten im Fahrzeug notwendig machen würde.

Die Aufgabe wird insbesondere ferner gelöst durch ein computerlesbares Speichermedium, welches Instruktionen enthält, die mindestens einen Prozessor dazu veranlassen, ein Verfahren, wie es vorstehend beschrieben ist, zu implementieren, wenn die Instruktionen durch den mindestens einen Prozessor ausgeführt werden.

Es ergeben sich ähnliche oder identische Vorteile, wie sie bereits im Zusammenhang mit dem Verfahren beschrieben worden sind.

Die Aufgabe wird insbesondere ferner gelöst durch ein Steuergerät, das dazu ausgebildet ist, unter Verwendung einer nach dem vorstehend beschriebenen Verfahren erzeugten ersten Programmkomponente, einer zweiten Programmkomponente und einer Proxykomponente Funktionen eines angeschlossenen Fahrzeugs zu steuern.

Es ergeben sich ähnliche oder identische Vorteile, wie sie bereits im Zusammenhang mit dem Verfahren beschrieben worden sind.

Die Aufgabe wird ferner insbesondere gelöst durch ein System, umfassend:

- ein Fahrzeug mit einem Steuergerät;
- eine Rechneinheit, die dazu ausgebildet ist:
 - o mindestens eine erste Programmeinheit und mindestens eine zweite Programmeinheit zu empfangen;

- eine erste und eine zweite Proxydefinition zu erzeugen, wobei eine Proxydefinition einen Zugriff auf mindestens eine Funktion und/oder einen Speicherbereich einer Programmeinheit festlegt, wobei die erste Proxydefinition der ersten Programmeinheit und die zweite Proxydefinition der zweiten Programmeinheit zugeordnet ist;
- die erste Programmeinheit und die zweite Programmeinheit zur Erzeugung einer ersten Programmkomponente, einer zweiten Programmkomponente und einer ersten und einer zweiten Proxykomponente zu kompilieren;

wobei das Steuergerät dazu ausgebildet ist die erste Programmkomponente und die zweite Programmkomponente auszuführen, wobei die erste Programmkomponente mindestens eine Funktion der zweiten Programmkomponente unter Verwendung der ersten und der zweiten Proxykomponente aufruft und/oder verwendet.

Das System kann die vorstehend beschiedene Speichereinheit umfassen.

Es ergeben sich ähnliche oder identische Vorteile, wie sie bereits im Zusammenhang mit dem Verfahren beschrieben worden sind.

Das System kann einige oder alle der vorherig beschriebenen Verfahrensschritte implementieren.

Im Folgenden wird die Erfindung anhand von Ausführungsbeispielen näher erläutert. Dabei zeigen

Figur 1: eine schematische Darstellung eines Fahrzeugs;

Figur 2: eine schematische Darstellung von drei Rechenkernen eines Steuergeräts;

Figur 3: eine schematische Darstellung von drei Programmkomponenten und der Kommunikation zwischen diesen;

Figur 4: eine schematische Darstellung der Erzeugung von drei Programmkomponenten unter Verwendung eines Regelsatzes; und

Figur 5: eine schematische Darstellung der Aktualisierung einer Programmkomponente.

Im Folgenden werden für gleiche oder gleichwirkende Teile dieselben Bezugsziffern verwendet.

Figur 1 zeigt ein Fahrzeug 1 mit einem Steuergerät 2 und einem Fahrerassistenzsystem 3 und einem Querdynamiksystem 4. Bei dem Fahrerassistenzsystem 3 handelt es sich in dem vorliegenden Ausführungsbeispiel um eine Bremsautomatik, die das Fahrzeug 1 zum stehen bringt, wenn ein vorrausfahrendes Fahrzeug abrupt bremst. Bei dem Fahrerassistenzsystem 3 handelt es sich also um ein sicherheitskritisches System.

Das Querdynamiksystem 4 steuert Funktionen, die im Wesentlichen die Querkräfte des Fahrzeugs 1 betreffen, wie z.B. die Lenkung.

Das Fahrerassistenzsystem 3 und das Querdynamiksystem 4 sind mit einem Steuergerät 2 kommunikativ verbunden. Beispielsweise können das Steuergerät 2, das Fahrerassistenzsystem 3 und das Querdynamiksystem 4 über einen Fahrzeugbus, z.B. einen CAN-BUS, miteinander verbunden sein.

Bei dem Steuergerät 2 handelt es sich in dem gezeigten Ausführungsbeispiel um ein Universalsteuergerät 2, das eine Vielzahl von Prozessoren und eine Speichereinrichtung aufweist. In der Speichereinrichtung sind drei Partitionen vorgesehen, wobei eine Betriebssystempartition ein Autosar OS-basiertes Betriebssystem speichert. Eine Fahrerassistenzpartition speichert Programmkomponenten zur Implementierung der gewünschten Fahrerassistenzfunktion. In diesem Ausführungsbeispiel also der Bremsautomatik. Eine Querdynamikpartition speichert Programmkomponenten zur Implementierung der Querdynamik.

Wenn eine neue Version der auf dem Steuergerät 2 ausgeführten Programmkomponenten installiert werden soll, wird das Steuergerät mit einem Entwicklungsrechner 5 drahtgebunden oder drahtlos verbunden. Bei dem

Entwicklungsrechner 5 kann es sich auch um einen sogenannten Tester oder eine sonstige Einrichtung handeln, mit der Programmkomponenten auf das Steuergerät 2 aufgespielt werden können. In einem Ausführungsbeispiel ist der Entwicklungsrechner 5 als ein mobiles Endgerät ausgebildet, das drahtlos mit dem Steuergerät 2 verbunden ist.

Es ist auch möglich, dass das Steuergerät über ein Mobilfunk- oder WLAN-Modul verfügt, das kommunikativ mit dem Entwicklungsrechner 5 verbindbar ausgebildet ist.

Die Figur 2 zeigt den logischen Aufbau des Steuergeräts 2 im Detail. Das Steuergerät 2 umfasst drei virtuelle Ausführungsumgebungen 10, 20, 30 zur Ausführung von Programmkomponenten. Jeder der Ausführungsumgebungen 10, 20, 30 sind jeweils Rechenkern eines Prozessors des Steuergeräts 2 zugewiesen. So weist eine erste virtuelle Ausführungsumgebung 10 einen ersten Rechenkern 11 und einen zweiten Rechenkern 12 auf. Eine zweite virtuelle Ausführungsumgebung 20 weist einen dritten Rechenkern 21 und einen vierten Rechenkern 22 auf. Eine dritte virtuelle Ausführungsumgebung 30 weist einen fünften Rechenkern 31 und einen sechsten Rechenkern 32 auf.

Die durch die entsprechenden Rechenkern 11, 12, 21, 22, 31, 32 ausgeführten Programmkomponenten können über eine Hardwareebene 7 miteinander kommunizieren. Das bedeutet, die Hardwareebene 7 stellt Einrichtungen bereit, die es einer Programmkomponente, die durch den ersten Rechenkern 11 ausgeführt wird, erlauben eine Funktion einer Programmkomponente, die durch den dritten Rechenkern 21 ausgeführt wird, aufzurufen.

Die Figur 3 illustriert die Kommunikation zwischen den Programmkomponenten, die von unterschiedlichen Rechenkernen ausgeführt werden. Hierzu zeigt die Figur 3 die erste, die zweite und die dritte virtuelle Ausführungsumgebung 10, 20, 30. In dem Ausführungsbeispiel der Figur 3 kann in der ersten virtuellen Ausführungsumgebung 10 das Betriebssystem des Steuergeräts 2 ausgeführt werden. Die zweite virtuelle Ausführungsumgebung 20 kann in dem Ausführungsbeispiel ein Fahrerassistenzsystem durch Programmkomponenten 23, 24 ausführen. Die dritte virtuelle Ausführungsumgebung 30 kann in dem Ausführungsbeispiel die Querdynamik des Fahrzeugs 1 durch die Programmkomponenten 33, 34 steuern.

Wie ausgeführt, kann es sich bei dem Fahrerassistenzsystem z.B. um eine automatische Bremsautomatik handeln. Beim Bremsen durch die Bremsautomatik kann es notwendig sein, dass auch die Querdynamik von der Bremsautomatik gesteuert wird. Das bedeutet, die Bremsautomatik muss auf Funktionen der Querdynamik zugreifen. Zum Beispiel kann die Bremsautomatik eine Funktion, die durch die Programmkomponente 34 bereitgestellt wird, aufrufen.

Für einen solchen Funktionsaufruf sendet die Rechenkomponente 23 der ersten virtuellen Ausführungsumgebung 10 ein Signal S, welches über eine Proxyschnittstelle 25 an eine weitere Proxys-Schnittstelle 13 der ersten virtuellen Ausführungsumgebung 10 zu einer internen Kommunikationseinheit 14 weitergeleitet wird. Von der internen Kommunikationseinheit 14 aus wird das Signal S zu einer Proxy-Schnittstelle 15 geleitet, die wiederum mit einer Proxy-Schnittstelle 35 der dritten virtuellen Ausführungsumgebung 30 kommunikativ verbunden ist. Die Proxy-Schnittstelle 35 initiiert dann den Funktionsaufruf unter Verwendung des Signals S in der Programmkomponente 34.

Die Figuren 4 und 5 illustrieren das erstmalige Kompilieren der eben beschriebenen Komponenten und das Aktualisieren einer einzelnen Programmkomponente.

Die Figur 4 zeigt zwei Programmeinheiten 8, 9, die in dem gezeigten Ausführungsbeispiel Quellcodedateien embedded-C darstellen. Zum Erstellen von ausführbaren Programmkomponenten ist es notwendig den Quellcode zu kompilieren und unterschiedliche Programmkomponenten zu verlinken. Dazu wird eine Linker-Compilereinheit 40 verwendet. Die Linker-Compilereinheit 40 umfasst eine Proxy-Erzeugungseinheit 41 und einen Linker 42. Die Proxy-Erzeugungseinheit 41 ist dazu ausgebildet, Funktionsaufrufe und Zugriffe zwischen den Programmeinheiten 8, 9 zu identifizieren. Aus diesen Informationen generiert die Proxy-Erzeugungseinheit 41 Proxykomponenten 13, 25. Darüber hinaus werden die durch ein Steuergerät 2 ausführbaren Programmkomponenten 23, 24 generiert.

Die Figur 5 zeigt einen Fall, in dem die Programmeinheit 9 aktualisiert wird. Hierzu empfängt die Linker-Compilereinheit 40 eine aktualisierte Programmeinheit 9', bei der die Implementierung einzelner Funktionen optimiert wurde. Die Schnittstellen und die Zugriffe auf andere Funktionen sind gleichgeblieben. Die

Linker-Compilereinheit 40, dort insbesondere der Linker 42, analysiert die Programmeinheit 9' und vergleicht diese mit der bestehenden Programmeinheit 9.

Eine Regelspeichereinheit 50 speichert Regeln R1-R4, die angeben, unter welchen Voraussetzungen es ausreicht, dass nur die Programmeinheit 9' kompiliert wird oder ob alle Programmeinheiten 8, 9' erneut kompiliert werden müssen.

So gibt die Regel R1 beispielsweise an, dass alle Programmeinheiten 8, 9' kompiliert werden müssen, wenn die Programmeinheit 9' die Größe von gemeinsam genutzten Speicherbereichen – sog. shared memory – ändert.

Unter Verwendung der Regeln R1-R4 und/oder der schon existierenden Programmkomponenten 23, 24 und/oder der existierenden Proxykomponenten 25, 13 analysiert die Linker-Compilereinheit 40, ob nur die Programmeinheit 9' kompiliert werden muss. In dem gezeigten Ausführungsbeispiel bestimmt die Linker-Compilereinheit 40, dass nur die Programmeinheit 9' kompiliert werden muss, sodass eine neue Programmkomponente 24' erzeugt wird, die die bisherige Programmkomponente 24 ersetzt. Dabei ist die Programmkomponente 24' dazu ausgebildet, die Proxykomponente 25 zum Zugriff auf Funktionen und Speicherbereiche der Programmkomponente 23 zu verwenden.

Insgesamt muss nur eine einzige Programmeinheit neu kompiliert werden. Dies reduziert die aufzubringende Rechenleistung, sodass das Programm schneller aktualisiert werden kann. Darüber hinaus ist es nicht notwendig, eine erneute Testfahrt mit einem Testfahrzeug durchzuführen, sofern es sich bei der neu kompilierten Komponente nicht um eine sicherheitskritische Komponente handelt. Dadurch können beachtliche Kosten eingespart werden.

Bezugszeichenliste:

1	Fahrzeug
2	Steuergerät
3	Fahrerassistenzsystem
4	Querdynamiksystem
5	Entwicklungsrechner
6	Kompilat/Binärdateien
7	Hardwareebene

8, 9, 9'	Programmeinheit
10	erste virtuelle Ausführungsumgebung
11	erster Rechenkern
12	zweiter Rechenkern
13, 15, 25, 35	Proxy-Schnittstelle
14	interne Kommunikationseinheit
20	zweite virtuelle Ausführungsumgebung
21	dritter Rechenkern
22	vierter Rechenkern
23, 24, 24', 33, 34	Programmkomponente
30	dritte virtuelle Ausführungsumgebung
31	fünfter Rechenkern
32	sechster Rechenkern
40	Linker-Compilereinheit
41	Proxyerzeugungseinheit
42	Linker
50	Regelspeichereinheit
R1-R4	Regel
S	Signal

Ansprüche

1. Verfahren zum Ausführen von Programmkomponenten (23, 24, 33, 34) auf einem Steuergerät (2), insbesondere auf einem Steuergerät (2) eines Fahrzeugs (1), bevorzugt eines PKWs (1), umfassend die folgenden Schritte:
 - Empfangen mindestens einer ersten Programmeinheit (8) und mindestens einer zweiten Programmeinheit (9, 9');
 - Erzeugen einer ersten und einer zweiten Proxydefinition, wobei eine Proxydefinition einen Zugriff auf mindestens einer Funktion und/oder einen Speicherbereich einer Programmeinheit (8, 9, 9') festlegt, wobei die erste Proxydefinition der ersten Programmeinheit (8) und die zweite Proxydefinition (9) der zweiten Programmeinheit (9, 9') zugeordnet ist;
 - Kompilieren der ersten Programmeinheit (8) und der zweiten Programmeinheit (9, 9') zur Erzeugung einer ersten Programmkomponente (23) und Kompilieren einer zweiten Programmkomponente (24) und einer ersten und einer zweiten Proxykomponente (13, 25);
 - Ausführen der ersten Programmkomponente (23) und der zweiten Programmkomponente (24) auf einem Steuergerät (2), wobei die erste Programmkomponente (23) mindestens eine Funktion der zweiten Programmkomponente (24) unter Verwendung der ersten und der zweiten Proxykomponente (13, 25) aufruft und/oder verwendet.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die erste und die zweite Programmkomponente (23, 24) als Binärdateien ausgebildet sind, die jeweils durch einen Rechenkern (11, 12, 21, 22, 31, 32) eines Steuergeräts (2) ausführbar sind.
3. Verfahren einem der vorhergehenden Ansprüche dadurch gekennzeichnet, dass das Ausführen ein Ausführen der ersten Programmkomponente (23) und/oder der ersten Proxykomponente (25) auf einem ersten Rechenkern (11, 12, 21, 22, 31, 32) des Steuergeräts (2) und ein Ausführen der zweiten Programmkomponente (24) und/oder der zweiten

Proxykomponente (13) auf einem zweiten Rechenkern (11, 12, 21, 22, 31, 32) des Steuergeräts umfasst.

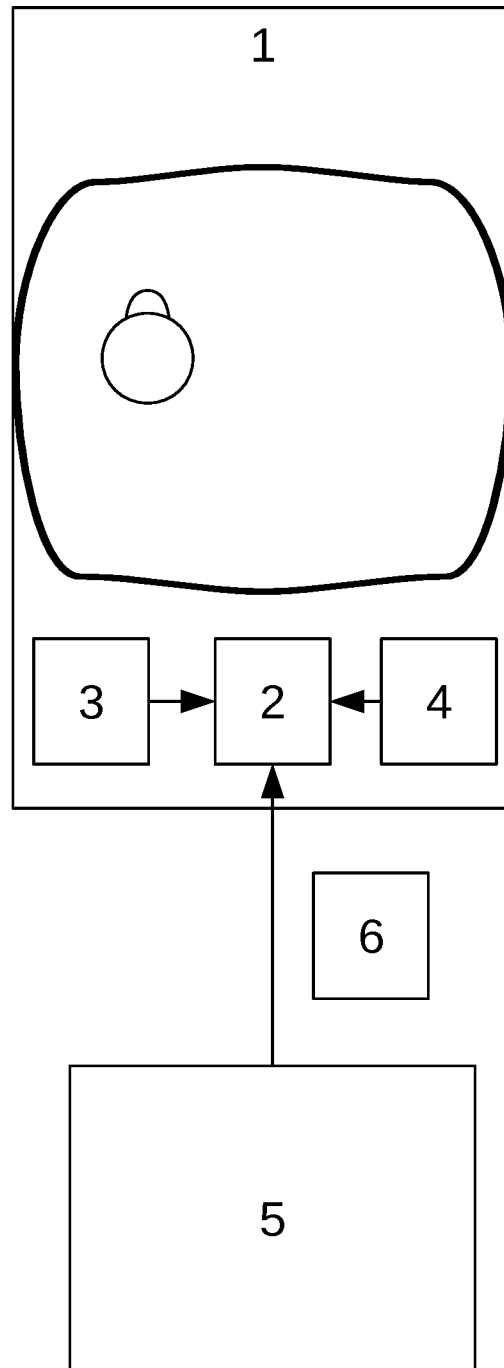
4. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die erste und/oder die zweite Programmeinheit (8, 9) Quellcode einer Programmiersprache, insbesondere embedded C, umfassen.
5. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die erste und/oder die zweite Proxydefinition Adressen von Funktionen der ersten bzw. zweiten Programmkomponente (8, 9) angeben.
6. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die erste und/oder die zweite Proxydefinition gemeinsam genutzte Speicherbereiche der ersten bzw. zweiten Programmkomponente (8, 9) angeben.
7. Verfahren nach einem der vorhergehenden Ansprüche, gekennzeichnet durch
 - Empfangen einer dritten Programmeinheit (9'), die hinsichtlich ihrer Schnittstellen der zweiten Programmeinheit (9) entspricht;
 - Kompilieren nur der dritten Programmeinheit (9') zur Erzeugung einer dritten Programmkomponente (24');
 - Ausführen der ersten Programmkomponente (23) und der dritten Programmkomponente (24') auf einem Steuergerät (2), wobei die dritte Programmkomponente (24') mindestens eine Funktion der ersten Programmkomponente (23) unter Verwendung der ersten und der zweiten Proxykomponente (25, 13) aufruft und/oder verwendet.
8. Verfahren nach einem der vorhergehenden Ansprüche, insbesondere nach Anspruch 7, dadurch gekennzeichnet, dass das Kompilieren der dritten Programmeinheit (9') unter Verwendung eines Regelsatzes ausgeführt wird, wobei der Regelsatz mindestens eine Regel (R1, R2, R3, R4) umfasst, die angibt, ob für ein Kompilieren der dritten

Programmeinheit (9') ein Kompilieren der ersten Programmeinheit (8) notwendig ist.

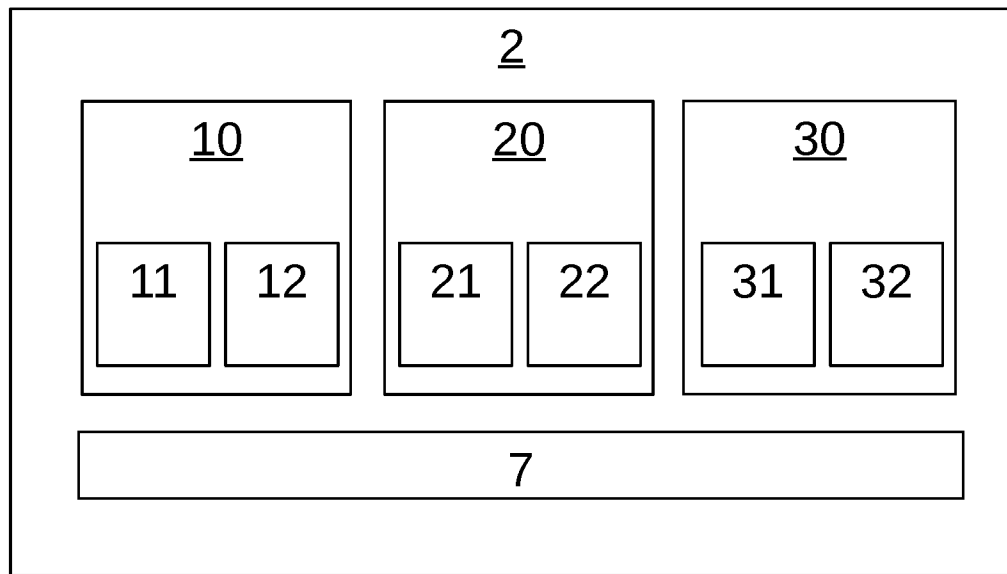
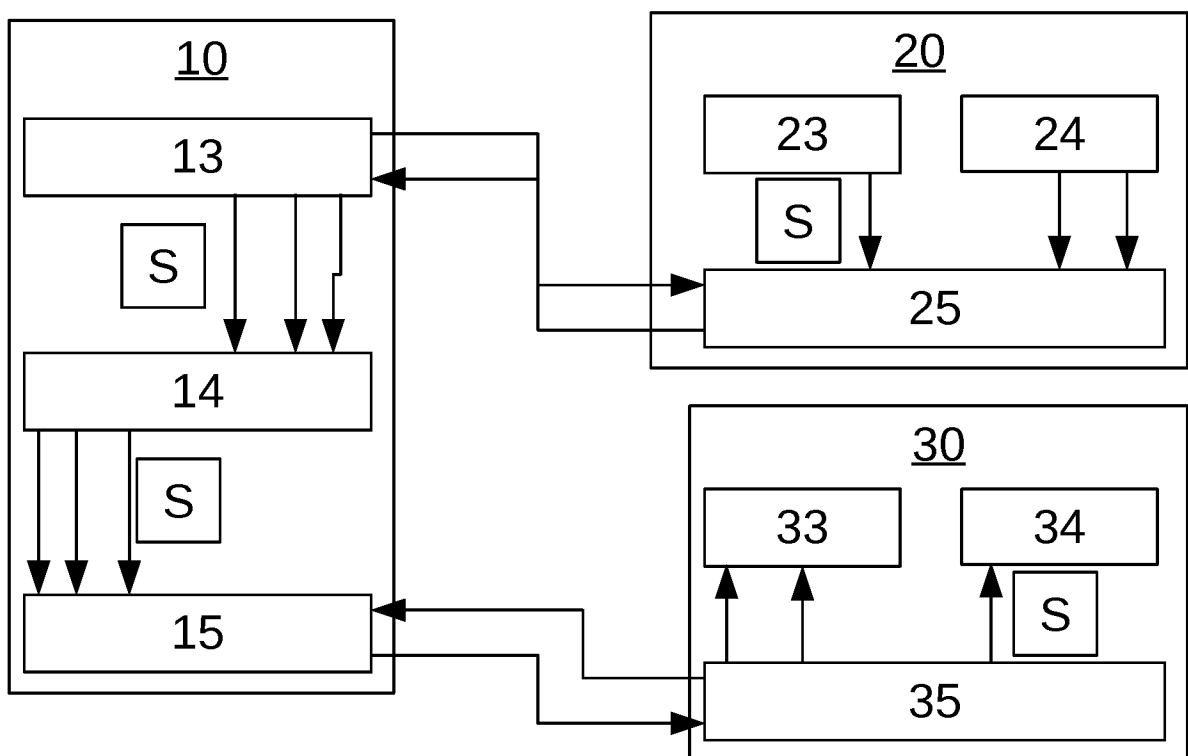
9. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die erste Programmkomponente (23) ein Betriebssystem für das Steuergerät (2) umfasst und/oder die erste Komponente (23) dazu ausgebildet ist, eine Querdynamik eines Fahrzeugs (1) zu steuern und/oder die zweite und/oder die dritte Programmkomponente (24, 24') dazu ausgebildet ist, mindestens ein Fahrerassistenzsystem eines Fahrzeugs (1) zu steuern.
10. Computerlesbare-Speichermedium, welches Instruktionen enthält, die mindestens einen Prozessor dazu veranlassen, ein Verfahren nach einem der vorhergehenden Ansprüche zu implementieren, wenn die Instruktionen durch den mindestens einen Prozessor ausgeführt werden.
11. Steuergerät (2), das dazu ausgebildet ist, unter Verwendung einer nach Anspruch 1 erzeugten ersten Programmkomponente (23), einer zweiten Programmkomponente (24) und mindestens einer Proxykomponente (13, 25) Funktionen eines angeschlossenen Fahrzeugs (1) zu steuern.
12. System, umfassend:
 - ein Fahrzeug (1) mit einem Steuergerät (2);
 - eine Rechneinheit, die dazu ausgebildet ist:
 - o mindestens eine erste Programmeinheit (8) und mindestens eine zweite Programmeinheit (9, 9') zu empfangen;
 - o eine erste und eine zweite Proxydefinition zu erzeugen, wobei eine Proxydefinition einen Zugriff auf mindestens einer Funktion und/oder einen Speicherbereich einer Programmeinheit festlegt, wobei die erste Proxydefinition der ersten Programmeinheit (8) und die zweite Proxydefinition der zweiten Programmeinheit (9, 9') zugeordnet ist;
 - o die erste Programmeinheit (8) und die zweite Programmeinheit (9, 9') zur Erzeugung einer ersten Programmkomponente (23), einer zweiten Programmkomponente (24, 24') und einer ersten und einer zweiten Proxykomponente (13, 25) zu kompilieren;

wobei das Steuergerät (2) dazu ausgebildet ist die erste Programmkomponente (23) und die zweite Programmkomponente (24) auszuführen, wobei die erste Programmkomponente (23) mindestens eine Funktion der zweiten Programmkomponente (24) unter Verwendung der ersten und der zweiten Proxykomponente (13, 25) aufruft und/oder verwendet.

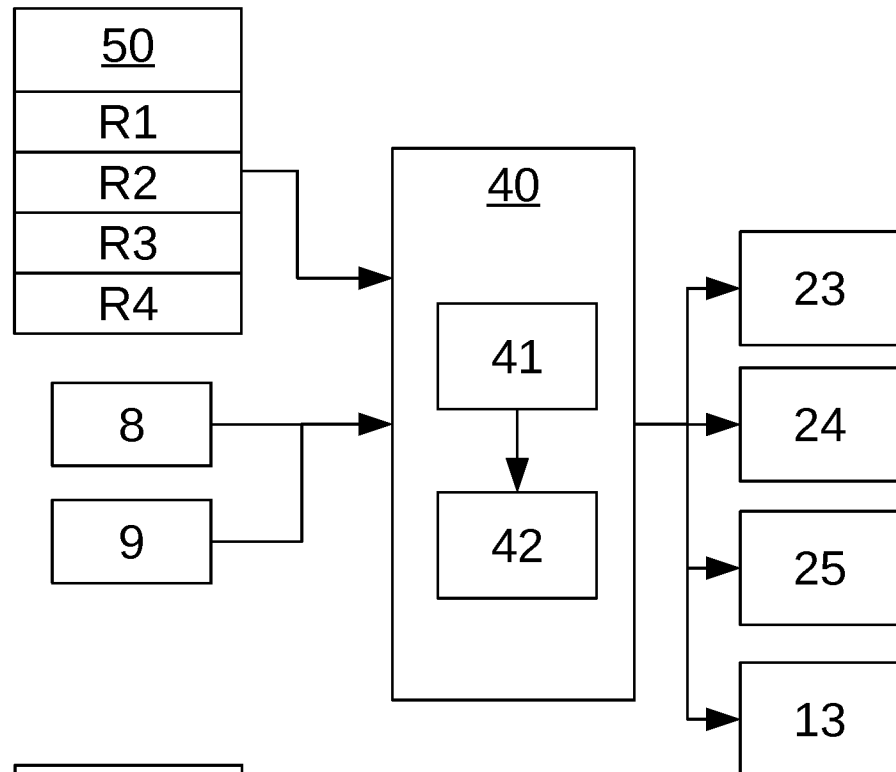
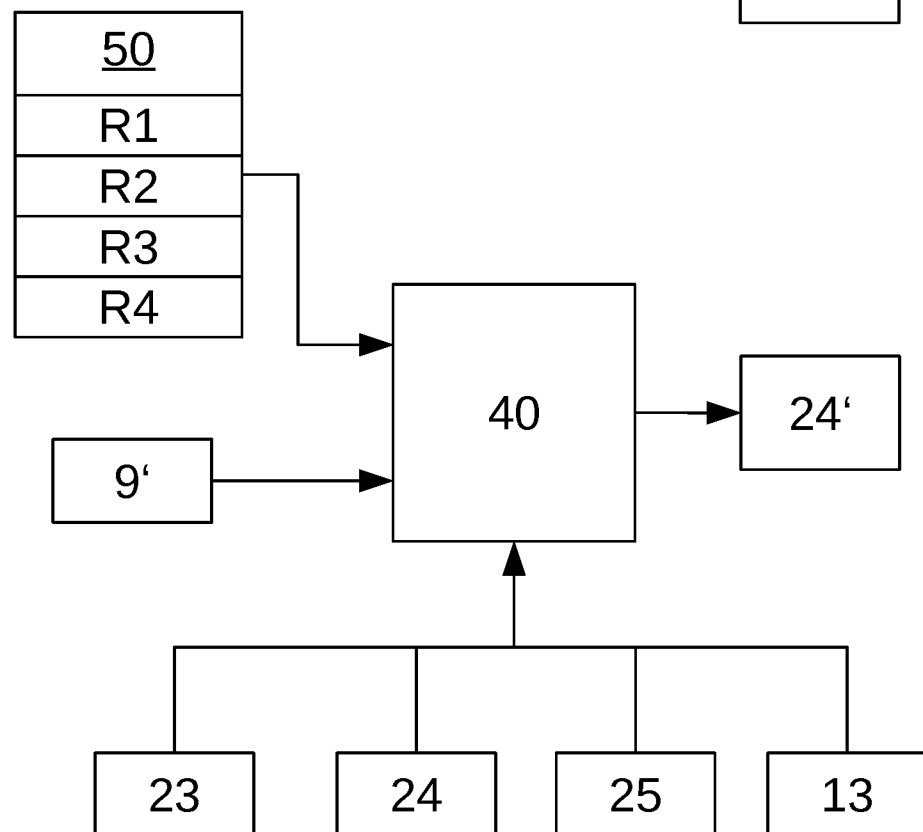
1 / 3

Fig. 1

2 / 3

Fig. 2Fig. 3

3 / 3

Fig. 4Fig. 5

INTERNATIONAL SEARCH REPORT

International application No.

PCT/EP2019/052315

A. CLASSIFICATION OF SUBJECT MATTER**G06F 8/30**(2018.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Leo Bernhard Geissmann ET AL. "Separate Compilation in Modula-2 and the Structure of the Modula-2 Compiler on the Personal Computer Lilith" , Zurich, 01 January 1983 (1983-01-01), pages 1-126, Retrieved from the Internet: https://www.cfbsoftware.com/modula2/ETH7286.pdf [retrieved on 2019-10-07] XP055629430 page 9, line 1 - page 12, line 27; figures 1.1, 1.2	1-12



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

08 October 2019

Date of mailing of the international search report

23 October 2019

Name and mailing address of the ISA/EP

European Patent Office
p.b. 5818, Patentlaan 2, 2280 HV Rijswijk
Netherlands

Telephone No. (+31-70)340-2040

Facsimile No. (+31-70)340-3016

Authorized officer

Müller, Tobias

Telephone No.

A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES

INV. G06F8/30

ADD.

Nach der Internationalen Patentklassifikation (IPC) oder nach der nationalen Klassifikation und der IPC

B. RECHERCHIERTE GEBIETE

Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole)

G06F

Recherchierte, aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen

Während der internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe)

EPO-Internal

C. ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
X	Leo Bernhard Geissmann ET AL: "Separate Compilation in Modula-2 and the Structure of the Modula-2 Compiler on the Personal Computer Lilith", 1. Januar 1983 (1983-01-01), Seiten 1-126, XP055629430, Zurich Gefunden im Internet: URL: https://www.cfbsoftware.com/modula2/ETH7286.pdf [gefunden am 2019-10-07] Seite 9, Zeile 1 - Seite 12, Zeile 27; Abbildungen 1.1, 1.2 -----	1-12



Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen



Siehe Anhang Patentfamilie

* Besondere Kategorien von angegebenen Veröffentlichungen :

"A" Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besonders bedeutsam anzusehen ist

"E" frühere Anmeldung oder Patent, die bzw. das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist

"L" Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt)

"O" Veröffentlichung, die sich auf eine mündliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Maßnahmen bezieht

"P" Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist

"T" Spätere Veröffentlichung, die nach dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondern nur zum Verständnis des der Erfindung zugrundeliegenden Prinzips oder der ihr zugrundeliegenden Theorie angegeben ist

"X" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung nicht als neu oder auf erfinderischer Tätigkeit beruhend betrachtet werden

"Y" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann nicht als auf erfinderischer Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit einer oder mehreren Veröffentlichungen dieser Kategorie in Verbindung gebracht wird und diese Verbindung für einen Fachmann naheliegend ist

"&" Veröffentlichung, die Mitglied derselben Patentfamilie ist

Datum des Abschlusses der internationalen Recherche

8. Oktober 2019

Absendedatum des internationalen Recherchenberichts

23/10/2019

Name und Postanschrift der Internationalen Recherchenbehörde

Europäisches Patentamt, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Bevollmächtigter Bediensteter

Müller, Tobias