



(10) **DE 10 2015 117 947 A1** 2017.04.27

(12)

Offenlegungsschrift

(21) Aktenzeichen: **10 2015 117 947.4**

(22) Anmeldetag: **21.10.2015**

(43) Offenlegungstag: **27.04.2017**

(51) Int Cl.: **H04L 12/24 (2006.01)**

(71) Anmelder:

**Sontheim Industrie Elektronik GmbH, 87437
Kempten, DE**

(74) Vertreter:

**Patentanwaltskanzlei Hutzelmann, 89296
Osterberg, DE**

(72) Erfinder:

**Sontheim, Bruno, 87437 Kempten, DE; Strobl,
Stephan, 87662 Kaltental, DE; Bottenbruch, Gerd,
87490 Haldenwang, DE; Lorenz, Martin, 87439
Kempten, DE**

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Verfahren zur Übermittlung von Daten**

(57) Zusammenfassung: Verfahren zur Übermittlung von Daten in einem verteilten System aus programmierbaren Steuerungen, wobei zwischen Datenquellen und Datensenken, die beide jeweils einen Knoten im verteilten System darstellen, Daten übertragen werden, wobei die jeweils miteinander kommunizierenden Datenquellen und Datensenken über Parameter bestimmt werden.

Beschreibung

[0001] Die Erfindung bezieht sich auf ein Verfahren zur Übermittlung von Daten in einem verteilten System aus programmierbaren Steuerungen.

[0002] Derartige Steuerungen kommunizieren miteinander über ein Netzwerk, Feldbus oder dergleichen.

[0003] Dabei wird in den Programmroutinen hinterlegt, welche Daten von einem Steuergerät an ein anderes gesendet oder von diesem empfangen werden sollen.

[0004] Auch die Adressierung des jeweils anderen Steuergerätes wird dabei in den Programmroutinen hinterlegt.

[0005] Dies ist jedoch mehr als problematisch, da selbst bei geringfügigen Änderungen, zum Beispiel durch Austausch eines Aggregates oder Steuergerätes diese Anpassung jedesmal im Programmcode vorgenommen werden muss.

[0006] Hierdurch muss dann der gesamte Programmcode jedesmal einer Evaluierung und verschiedenen Tests unterzogen werden, um sicherzustellen, daß Fehlfunktionen wenigstens weitgehend ausgeschlossen sind.

[0007] Aufgabe der Erfindung ist es, ein Verfahren vorzuschlagen, welches es ermöglicht, auf diese aufwendigen Evaluierungen und Tests, die auch erhebliche Kosten verursachen, zu verzichten.

[0008] Diese Aufgabe wird erfindungsgemäß dadurch gelöst, daß zwischen Datenquellen und Datensenken, die beide jeweils einen Knoten im verteilten System darstellen, Daten übertragen werden, wobei die jeweils miteinander kommunizierenden Datenquellen und Datensenken über Parameter bestimmt werden.

[0009] Damit wird das Netzwerk aus Datenquellen und Datensenken unabhängig von den hinterlegten Programmroutinen über Parameter definiert.

[0010] Dabei hat es sich als sehr vorteilhaft erwiesen, wenn die Knoten jeweils mit einem Steuerprogramm und Standardfunktionen ausgerüstet sind.

[0011] Hierdurch kann das Steuerprogramm von Standardfunktionen entkoppelt werden, auf die das Steuerprogramm zugreifen kann.

[0012] Äußerst vorteilhaft ist es in diesem Zusammenhang auch, wenn die Kommunikationsfunktionen als feste Programmroutinen hinterlegt sind.

[0013] Hierdurch wird dem Steuerprogramm Zugriff auf Kommunikationsfunktionen ermöglicht.

[0014] Dabi ist es gemäß einer weiteren Ausgestaltung der Erfindung sehr vorteilhaft, wenn die Programmroutinen für die Kommunikationsfunktionen parametrierbar sind.

[0015] Damit kann der Zugriff auf die Kommunikationsfunktionen durch Parameter angepasst werden.

[0016] Äußerst vorteilhaft ist es dabei auch, wenn die Parametrierung über eine externe Konfigurationsdatei erfolgt.

[0017] Damit lassen sich die Parameter unabhängig vom Steuerprogramm anpassen. Es ist sogar eine Anpassung während der Laufzeit des Steuerprogrammes denkbar.

[0018] Gemäß einer weiteren Ausgestaltung der Erfindung ist es äußerst vorteilhaft, wenn Start- und Zieladresse der Kommunikationsfunktionen parametrierbar sind.

[0019] Hierdurch wird eine sehr große Flexibilität gewährleistet. Es lassen sich problemlos Komponenten ggf. im laufenden Betrieb austauschen.

[0020] Weiterhin ist es sehr vorteilhaft, wenn die den Kommunikationsfunktionen zugrundeliegenden Protokolle parametrierbar sind.

[0021] Damit können die einzelnen Module oder zumindest Programmroutinen auch für verschiedenste Anwendungen eingesetzt werden und können beispielsweise auf Feldbus, Ethernet oder dergleichen aufbauen.

[0022] Gemäß einer Weiterbildung der Erfindung ist es auch äußerst vorteilhaft, wenn die Programmroutinen, Kommunikationsfunktionen, Hardware APIs usw. virtualisiert sind.

[0023] Sehr vorteilhaft ist es erfindungsgemäß auch, wenn die Programmroutinen, Kommunikationsfunktionen, Protokolle usw. hardwareunabhängig ausgeführt sind.

[0024] Dadurch wird weitgehende Hardwareunabhängigkeit erreicht. Die Programmroutinen werden universell einsetzbar und können auf verschiedener Hardware laufen. So kann beispielsweise ein vernetztes Fahrzeug anstatt mit einem Dieselmotor auch mit einem Hybridantrieb ausgerüstet werden. Der Antrieb ist jeweils mit einem spezifischen Steuergerät versehen, welches standardisierte Daten mit anderen Steuergeräten austauscht.

[0025] Im folgenden wird die Erfindung anhand eines Ausführungsbeispiels veranschaulicht.

[0026] Die Erfindung wird anhand der Steuerung einer Pistenraupe veranschaulicht, ist jedoch auf beliebige Steuerungs- und Regelsysteme übertragbar.

[0027] Im Fahrzeug sind die Steuerungen und Sensoren als Einheiten aufgebaut, die über ein Bussystem miteinander verbunden sind. Jeder Einheit bildet dabei einen Netzwerkknoten.

[0028] Knoten, die Daten erfassen und weitergeben sind dabei Datenquellen, solche, die Daten empfangen und weiterverarbeiten sind Datensenken.

[0029] Wird nun von einer Datenquelle beispielsweise die Motordrehzahl erfasst, so wird diese in einer Nachricht kodiert und versandt. Hierzu gibt der Programmcode der Datenquelle den Befehl, das Datenpaket zu versenden. Spezielle Kommunikationsfunktionen senden das Datenpaket an die Datensenke.

[0030] Dabei greifen die Kommunikationsfunktionen auf eine Parameterdatei zurück, in der spezifiziert ist, welches Datenpaket mit welchem Protokoll über welches Bussystem an welche Datensenke versandt werden soll.

[0031] Diese Parameterdatei ist dabei unabhängig vom Programmcode.

[0032] Der Programmcode wird beispielsweise in einer Laufzeitumgebung oder einer virtuellen Maschine ausgeführt und ist damit unabhängig von der verwendeten Hardware. Es können dabei standardisierte Laufzeitumgebungen wie beispielsweise Java eingesetzt werden. Es kann auch eine Abstraktion zwischen hardwareabhängigen Routinen und hardwareunabhängigen Routinen vorgenommen werden, so daß der eigentliche Programmcode hardwareunabhängig ist und auf hardwareabhängige Routinen zugreift. Gerade auf Embedded-Systemen ist dies besonders sinnvoll, da dort ein Betriebssystem in der Regel nicht vorgesehen ist.

[0033] Die Kommunikationsfunktionen stellen eine standardisierte Schnittstelle für den Programmcode zur Verfügung, auf die dieser zugreifen kann.

[0034] Die Kommunikationsfunktionen können hardware-spezifisch sein, oder aber über spezielle Hardwaretreiber auf die Hardware und damit die physikalischen Schnittstellen zugreifen.

[0035] Ebenso kann nicht nur die Kommunikation mit anderen Knoten derart abstarhiert werden, sondern es ist auch denkbar, daß Mikrokontroller, welche Sensoren auswerten und so als Datenquellen dienen, auf dieselbe Art und Weise auf die Sensoren zugreifen,

so daß auch hier Hardwareunabhängigkeit erreicht wird.

[0036] Die Datensenken verarbeiten die empfangenen Datenpakete. Um diese empfangen zu können, greift der dortige Programmcode ebenfalls auf Kommunikationsfunktionen zu, welche die Datenpakete empfangen und an den Programmcode übergeben.

[0037] Auch hier ist der Programmcode hardwareunabhängig, wohingegen die Kommunikationsfunktionen direkt oder indirekt auf die Hardware zugreifen.

[0038] Um bei dem Beispiel der Pistenraupe zu bleiben: Wird nun der Dieselmotor gegen ein Hybridagregat ausgetauscht, so muss nur die Parametrierung der Steuereinheit des Hybridagregats entsprechend Parametriert werden, so daß Drehzahlinformationen an eine Anzeige, welche durch eine auf einem Mikrokontroller laufende App dargestellt werden kann.

[0039] Der Anzeige Mikrokontroller „merkt“ dabei nicht einmal, daß ein anderes Antriebsagregat verbaut ist, da es seine einzige Aufgabe ist, die Drehzahlinformationen anzuzeigen.

[0040] Ebenso ist es denkbar, daß ein CAN Bus System auf Ethernet umgerüstet wird.

[0041] Dazu werden die Kommunikationsschnittstellen ausgetauscht und die Parametrierung der einzelnen Knoten entsprechend angepasst.

[0042] Der Programmcode kann unverändert übernommen werden.

[0043] Eine erneute Evaluierung und umfangreiche Tests entfallen.

[0044] Auch kann die Parametrierung im laufenden Betrieb geändert werden. Dies ist mit herkömmlichen Ansätzen, bei denen die Kommunikation direkt im Programmcode hinterlegt ist, nicht möglich.

[0045] Vielmehr muss bei jeder noch so kleinen Änderung der Code komplett neu kompiliert und getestet werden.

[0046] Mit dem erfindungsgemäßen Ansatz wird zudem erreicht, daß Programmcode oftmals für verschiedenste Anwendungen eingesetzt werden kann, wodurch die Zuverlässigkeit von Geräten und Anlagen aufgrund langer und weitreichender Erfahrungen mit dem jeweiligen Programmcode erhöht wird.

Patentansprüche

1. Verfahren zur Übermittlung von Daten in einem verteilten System aus programmierbaren Steuerungen, **dadurch gekennzeichnet**, daß zwischen Da-

tenquellen und Datensinken, die beide jeweils einen Knoten im verteilten System darstellen, Daten übertragen werden, wobei die jeweils miteinander kommunizierenden Datenquellen und Datensinken über Parameter bestimmt werden.

2. Verfahren nach Anspruch 1, **dadurch gekennzeichnet**, daß die Knoten jeweils mit einem Steuerprogramm und Standardfunktionen ausgerüstet sind.

3. Verfahren nach Anspruch 1 oder 2, **dadurch gekennzeichnet**, daß die Kommunikationsfunktionen als feste Programmroutinen hinterlegt sind.

4. Verfahren nach Anspruch 3, **dadurch gekennzeichnet**, daß die Programmroutinen für die Kommunikationsfunktionen parametrierbar sind.

5. Verfahren nach Anspruch 4, **dadurch gekennzeichnet**, daß die Parametrierung über eine externe Konfigurationsdatei erfolgt.

6. Verfahren nach einem der vorangehenden Ansprüche, **dadurch gekennzeichnet**, daß Start- und Zieladresse der Kommunikationsfunktionen parametrierbar sind.

7. Verfahren nach einem der vorangehenden Ansprüche, **dadurch gekennzeichnet**, daß die den Kommunikationsfunktionen zugrundeliegenden Protokolle parametrierbar sind.

8. Verfahren nach einem der vorangehenden Ansprüche, **dadurch gekennzeichnet**, daß die Programmroutinen, Kommunikationsfunktionen, Hardware APIs usw. virtualisiert sind.

9. Verfahren nach einem der vorangehenden Ansprüche, **dadurch gekennzeichnet**, daß die Programmroutinen, Kommunikationsfunktionen, Protokolle usw. hardwareunabhängig ausgeführt sind.

Es folgen keine Zeichnungen