



(12) **Patentschrift**

(21) Aktenzeichen: **101 52 235.5**
(22) Anmeldetag: **20.10.2001**
(43) Offenlegungstag: **30.04.2003**
(45) Veröffentlichungstag
der Patenterteilung: **08.01.2015**

(51) Int Cl.: **H04L 1/24 (2006.01)**
H04L 12/407 (2006.01)
H03M 13/09 (2006.01)

Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(73) Patentinhaber:
Robert Bosch GmbH, 70469 Stuttgart, DE

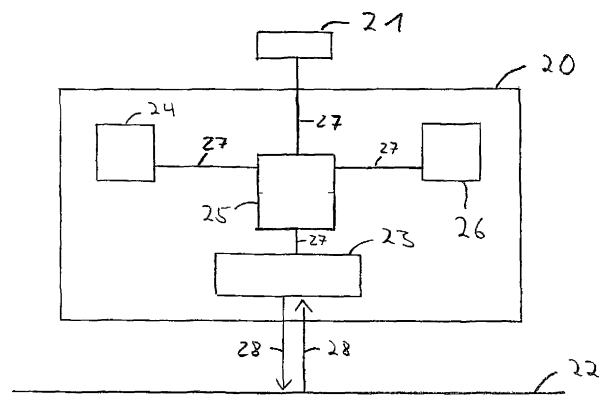
(72) Erfinder:
Emde, Christoph, 74211 Leingarten, DE;
Hesselbarth, Johannes, 71696 Möglingen, DE;
Springer, Alexander, 74376 Gemrigheim, DE

(56) Ermittelter Stand der Technik:

DE	691 14 380	T2
US	5 459 850	A
US	5 323 385	A

(54) Bezeichnung: **Verfahren zum Erkennen von Fehlern bei der Datenübertragung innerhalb eines CAN-Controllers und ein CAN-Controller zur Durchführung dieses Verfahrens**

(57) Hauptanspruch: Verfahren zum Erkennen von Fehlern bei der Datenübertragung innerhalb eines CAN-Controllers (13, 20), insbesondere bei der Datenübertragung zwischen einem Mikroprozessor (14, 21) und einer Schnittstelleneinheit (23), wobei für über einen CAN-Bus (11, 22) zu sendende Daten von dem Mikroprozessor (14, 21) zumindest ein Prüfbit erzeugt wird, das zur Sicherstellung der Konsistenz der übertragenen Daten überprüft werden kann, dadurch gekennzeichnet, dass das erzeugte Prüfbit der Schnittstelleneinheit (23) anzeigt, ob sicherheitsrelevante Daten vorliegen, wobei die Schnittstelleneinheit (23) das mindestens eine Prüfbit bei sicherheitsrelevanten Daten überprüft.



Beschreibung

[0001] Die Erfindung betrifft ein Verfahren zum Erkennen von Fehlern bei der Datenübertragung innerhalb eines CAN-Controllers und einen CAN-Controller zur Durchführung dieses Verfahrens.

Stand der Technik

[0002] In den letzten Jahren ist bei Kraftfahrzeugen ein deutlicher Trend hin zu immer stärkeren Einsatz von Steuerungs- und Regelungstechnik zu verzeichnen. Es gibt Motor- und Getriebesteuerungen, Antiblockiersysteme (ABS), Antischlupfregelungen (ASR) und mehrere Funktionen umfassende Fahrdynamikregelungen. Um einen Datenaustausch unter Echtzeitbedingungen zwischen diesen Komponenten zu ermöglichen, sind leistungsfähige Bus-Systeme erforderlich. Aus diesem Grunde wurde ein für diese Aufgaben optimiertes Bus-System, das Controller Area Network (CAN) entwickelt.

[0003] Aber auch über den Fahrzeugbereich hinaus ist CAN in anderen industriellen Anwendungen weit verbreitet.

[0004] Der CAN-Bus ist ein serieller Bus, an den die einzelnen Stationen, bspw. Steuergeräte, über CAN-Kontroll-Einheiten (CAN-Controller) angeschlossen sind. Die einzelnen Stationen haben die Möglichkeit, über den Bus durch Senden und Empfangen von Nachrichten mit den anderen Stationen zu kommunizieren.

[0005] Bei Kraftfahrzeugen sind viele der übertragenen Daten sicherheitsrelevant. Das heißt, daß diese Daten zur Steuerung der im Kraftfahrzeug befindlichen sicherheitsrelevanten Systeme, wie beispielsweise des Bremssystems, dienen. Bei sicherheitsrelevanten Systemen muß sichergestellt sein, daß sich auch im Fehlerfall kein sicherheitskritischer Zustand einstellt. Es muß also gewährleistet sein, daß sicherheitsrelevante Daten bei der Übertragung nicht verfälscht werden oder aber eine Verfälschung erkannt wird.

[0006] Bei jeder Übertragung einer Nachricht kann es – wenn auch mit geringer Wahrscheinlichkeit – geschehen, daß ein Bit innerhalb der Nachricht seinen Wert ändert, also bspw. aus einer 1 eine 0 wird. Es kann sogar vorkommen, daß gleich mehrere Bits innerhalb einer Nachricht "umkippen".

[0007] Die Hauptursache für solche Störungen sind elektromagnetische Fremdfelder, die in technischen Anlagen nicht zu vermeiden sind.

[0008] Da das Entstehen eines Fehlers, also das "Umkippen" eines Bits, nicht verhindert werden kann, muß man versuchen, den Fehler sicher zu erkennen.

Dies geschieht durch Einfügen redundanter Prüfbits in die Nachricht durch den Sender. Der Empfänger kann anhand der Prüfbits feststellen, ob die Übertragung korrekt oder fehlerhaft war.

[0009] CAN-Controller unterstützen generell eine sehr zuverlässige Datenübertragung. Die Daten werden beim Senden mit CRC-Prüfbits versehen. Diese werden beim Empfang überprüft. Auf diese Weise kann eine fehlerhafte Übertragung erkannt und entsprechend reagiert werden. So können zum Beispiel die zuvor nicht korrekt empfangenen Daten nochmals angefordert werden.

[0010] Die eigentliche Schnittstelle in einem CAN-Controller zu einem CAN-Bus wird als CAN-Protocol-Cernel bezeichnet. Mit diesem werden Daten zum Bus gesendet und vom Bus empfangen. Beim Senden erzeugt der CAN-Protocol-Cernel die Prüfbits und versendet diese mit den dazugehörigen Daten. Beim Empfangen einer Nachricht überprüft der CAN-Protocol-Cernel die empfangenen Prüfbits und stellt so fest, ob die Daten fehlerfrei übermittelt wurden.

[0011] Mit den CRC-Prüfbits im CAN wird aber nur die besonders störanfällige Übertragung zwischen den Controllern überwacht. Aber auch bei der Übertragung innerhalb des CAN-Controllers sowie bei der Übertragung zwischen CAN-Controller und einem Mikroprozessor, der zentralen Recheneinheit (CPU), können Übertragungsfehler nicht ausgeschlossen werden. Bei herkömmlichen CAN-Controllern besteht daher eine Lücke in der Fehlerüberwachung der übertragenen Daten.

[0012] Zur Absicherung der internen Datenübertragung wird in herkömmlichen Systemen zum Beispiel ein sogenannter "Loop Back Mode" verwendet. Dabei werden der CAN-Bus-seitige Ausgang und der Eingang des CAN-Protocol-Cernel verbunden und definierte Daten von der CPU gesendet und direkt wieder empfangen. Durch Vergleich der gesendeten und empfangenen Daten kann die gesamte interne Übertragungsstrecke geprüft werden.

[0013] Nachteilig bei diesem Verfahren ist, daß das Steuergerät während der Prüfung vom Bus abzukoppeln ist, damit die Prüfdaten am Bus nicht sichtbar werden. In vielen Systemen ist eine zeitweilige Abkopplung eines Busteilnehmers nicht akzeptabel, so daß ein "Loop Back Test" nur einmalig beim Einschalten durchgeführt werden kann.

[0014] Das vorstehend beschriebene Verfahren erlaubt somit keine permanente Überwachung des gesamten Übertragungsweges. Dies ist jedoch bei sicherheitskritischen Systemen erforderlich. Hier setzt die Erfindung an.

[0015] Die Patentschrift DE 691 14 380 T2 offenbart ein Verfahren zum Erkennen von Fehlern bei der internen Datenübertragung innerhalb einer Speichervorrichtung, bei dem zumindest ein Prüfbit erzeugt wird, das zur Sicherstellung der Konsistenz der übertragenen Daten überprüft werden kann.

[0016] Die Patentschrift US 5 459 850 A offenbart eine Fehlererkennung anhand eines Prüfbits bei der Datenübertragung zwischen einem Mikroprozessor und einem Speicherelement.

[0017] Die Patentschrift US 5 323 385 A offenbart die Übertragung von Zusatzinformation zur Fehlererkennung in einem Teilabschnitt einer CAN-Nachricht über einen CAN-Bus.

[0018] Keines der Dokumente betrifft die spezifische Absicherung der Übertragung von sicherheitsrelevanten Daten.

[0019] Aufgabe der nachstehend offenbarten Erfindung ist es, zu gewährleisten, dass sicherheitsrelevante bei der Übertragung nicht verfälscht werden oder aber eine Verfälschung erkannt wird.

[0020] Diese Aufgabe wird gelöst durch das Verfahren gemäß Patentanspruch 1 und den CAN-Controller gemäß Patentanspruch 8. Die abhängigen Patentansprüche betreffen vorteilhafte Ausführungsformen der Erfindung.

Vorteile der Erfindung

[0021] Das erfindungsgemäße Verfahren zum Erkennen von Fehlern bei der Datenübertragung innerhalb eines CAN-Controllers, insbesondere bei der Datenübertragung zwischen einem Mikroprozessor und einer Schnittstelleneinheit, zeichnet sich dadurch aus, daß für über einen CAN-Bus zu sendende Daten von dem Mikroprozessor zumindest ein Prüfbit erzeugt wird, das zur Sicherstellung der Konsistenz der übertragenen Daten überprüft werden kann, wobei das erzeugte Prüfbit der Schnittstelleneinheit anzeigt, ob sicherheitsrelevante Daten vorliegen, wobei die Schnittstelleneinheit das mindestens eine Prüfbit bei sicherheitsrelevanten Daten überprüft.

[0022] Beim Empfangen von Daten umfaßt das Verfahren bevorzugt folgende Verfahrensschritte: Zunächst werden die auf dem CAN-Bus befindlichen Daten mit der Schnittstelleneinheit empfangen. Anschließend wird in dieser Schnittstelleneinheit für die empfangenen Daten zumindest ein Prüfbit ermittelt und erzeugt. Die Daten werden mit dem mindestens einen Prüfbit in eine Speichereinheit abgelegt. Anschließend werden die Daten mit dem mindestens einen Prüfbit von dem Mikroprozessor eingelesen. Dieser kann dann abschließend überprüfen, ob ein Fehler bei der Datenübertragung vorliegt.

[0023] Zur effektiven Übertragung nicht-kritischer Daten ist das Verfahren bevorzugt abschaltbar. Der Mikroprozessor stellt dann fest, ob die empfangenen Daten sicherheitsrelevant sind oder nicht. In Abhängigkeit hiervon erfolgt eine Überprüfung des mindestens einen Prüfbits oder nicht.

[0024] Die Datenübertragung in dem CAN-Controller erfolgt vorteilhafterweise über eine Busschnittstelle. Die Logik in der Busschnittstelle dient zur Übertragung der Daten, die dann entsprechend den Anforderungen der einzelnen Komponenten in der Busschnittstelle gewandelt werden.

[0025] Die Fehlererkennung ist umso wirksamer, je mehr redundante Prüfbits zum Prüfen übertragen werden. Prinzipiell kann man sogar so viele Prüfbits in ein Telegramm einfügen, daß der Empfänger damit die fehlerhaften Bits im Telegramm ermitteln und somit auch korrigieren kann. Mit steigender Prüfbitzahl nimmt allerdings die Übertragungseffizienz ab. In der Zeit, in der man Prüfbits über die Leitung schickt, können keine neuen Datenbits übertragen werden. Man muß auch hier einen Kompromiß zwischen Sicherheit und Effizienz der Übertragung schließen.

[0026] Eine einfache Form der Datensicherung stellt das Parity-Bit dar. Zu den Datenbits wird ein zusätzliches Bit, das Parity-Bit, so hinzugefügt, daß die Anzahl der Einsen innerhalb der Bits, also Datenbits plus Parity-Bit, gerade ist. Wenn auf dem Übertragungsweg ein Bit "umkippt", ist die Anzahl der Einsen bei den Datenbits plus Parity-Bit nicht mehr gerade. So stellt der Empfänger fest, daß ein Fehler vorliegt. Wenn aber gleichzeitig zwei oder eine andere gerade Anzahl Bits ihren Wert ändern, läßt sich dieser Fehler nicht mehr mit dem Parity-Bit feststellen.

[0027] Eine weitere Möglichkeit der Datensicherung stellt die Ermittlung einer Prüfsumme dar.

[0028] Ein sehr effizientes Verfahren der Datensicherung stellt der sogenannte Cyclic Redundancy Check (CRC) dar. Dieses Verfahren wird beim CAN-Bus auch bei der Datenübertragung zwischen den Stationen, die an den CAN-Bus angekoppelt sind, eingesetzt.

[0029] Bevorzugt umfaßt das Verfahren beim Senden von Daten folgende Verfahrensschritte: Zunächst ermittelt und erzeugt der Mikroprozessor zu den zu sendenden Daten, die vom Mikroprozessor zur Verfügung gestellt worden sind, zumindest ein Prüfbit. Dieses wird zusammen mit den zu übertragenden Daten in der Speichereinheit abgelegt. Die Daten werden mit den Prüfbits von der Schnittstelleneinheit ausgelesen. Diese kann dann anhand der Prüfbits überprüfen, ob ein Fehler bei der Datenübertragung vorliegt. Ist dies nicht der Fall, werden die Daten auf den CAN-Bus gegeben.

[0030] Bevorzugt kann der Mikroprozessor beim Senden entscheiden, ob eine sicherheitsrelevante Übertragung vorliegt und die Prüfbits ermitteln, oder aber ohne Rechenaufwand Prüfbits erzeugen, die der Schnittstelleneinheit signalisieren, daß eine Überprüfung der Datenübertragung nicht notwendig ist.

[0031] Der erfindungsgemäße CAN-Controller weist eine Schnittstelleneinheit für den Datenaustausch mit einem CAN-Bus, eine Speichereinheit, in welcher empfangene Daten und zu sendende Daten abgelegt sind, und eine elektronische Einheit zum Steuern der Datenübertragung zwischen der Speichereinheit und der Schnittstelleneinheit auf. Er zeichnet sich dadurch aus, daß die Schnittstelleneinheit Mittel zum Erzeugen von mindestens einem Prüfbit für empfangene Daten und zum Überprüfen von mindestens einem Prüfbit für zu sendende Daten aufweist.

[0032] Zur Erkennung von Fehlern bei der Datenübertragung innerhalb des CAN-Controllers wird somit mindestens ein weiteres internes Prüfbit eingeführt. Das Prüfbit bzw. die Prüfbits werden zusammen mit den Daten in der Speichereinheit abgelegt. Beim Empfangen der Daten werden die Prüfbits in der Schnittstelleneinheit erzeugt und können anschließend von einem Mikroprozessor, welcher die Daten aus der Speichereinheit einliest, überprüft werden. Beim Senden von Daten wird das mindestens eine Prüfbit in dem Mikroprozessor erzeugt und kann dann von der Schnittstelleneinheit geprüft werden, bevor diese die Daten auf den CAN-Bus gibt.

[0033] Die Mittel zum Erzeugen und Überprüfen des mindestens einen Prüfbits in der Schnittstelleneinheit können in Hardware implementiert sein oder als Computerprogramm vorliegen. Das Computerprogramm enthält dann Programmcode-Mittel zum Erzeugen und Überprüfen von mindestens einem Prüfbit.

[0034] Ein erfindungsgemäßes Computerprogrammprodukt weist die Mittel zum Erzeugen und Überprüfen von mindestens einem Prüfbit auf. Die Mittel bzw. Programmcode-Mittel sind auf einem computerlesbaren Datenträger gespeichert. Als geeignete Datenträger können EEPROMs, Flash Memories, aber auch CD-ROMs, Disketten oder Festplattenlaufwerke verwendet werden.

[0035] Bevorzugt ist die Speichereinheit im CAN-Controller ein RAM-Baustein, bei dem für jedes mögliche Datum ein bestimmter Speicherbereich vorgesehen ist. Dies erleichtert erheblich die Kommunikation zwischen den einzelnen Komponenten.

[0036] Vorteilhafterweise ist die elektronische Einheit im CAN-Controller, die zum Steuern der Datenübertragung dient, eine Zustandsmaschine. Zustandsmaschinen sind mit den gebräuchlichen Ent-

wicklungswerkzeugen leicht zu entwerfen und zu implementieren.

[0037] Von Vorteil ist es, wenn in dem CAN-Controller eine Busschnittstelle vorgesehen ist, deren interne Logik zur Datenübertragung zwischen den einzelnen Komponenten, nämlich der Schnittstelleneinheit, der Speichereinheit, der elektronischen Einheit und dem Mikroprozessor, dient.

[0038] Das erfindungsgemäße Steuergerät weist einen vorstehend beschriebenen CAN-Controller, einen Mikroprozessor und eine Speichereinrichtung auf. Der Mikroprozessor verfügt über Mittel zum Erzeugen und Überprüfen von mindestens einem Prüfbit.

[0039] Nach dem Empfangen der Daten wird in der Schnittstelleneinheit im CAN-Controller mindestens ein Prüfbit erzeugt. Die Daten werden zusammen mit dem Prüfbit bzw. mit den Prüfbits in der Speichereinheit abgelegt. Der Mikroprozessor liest die Daten mit dem mindestens einem Prüfbit aus und kann das mindestens eine Prüfbit überprüfen. Anhand des Prüfbits bzw. der Prüfbits kann der Mikroprozessor feststellen, ob ein Fehler bei der Datenübertragung innerhalb des CAN-Controllers vorliegt.

[0040] Beim Senden von Daten wird mindestens ein Prüfbit in dem Mikroprozessor ermittelt und erzeugt und anschließend mit den Daten in der Speichereinheit abgelegt. Die Schnittstelleneinheit, welche die zu sendenden Daten aus der Speichereinheit ausliest, kann das mindestens eine Prüfbit überprüfen und so feststellen, ob ein Fehler in der Datenübertragung vorliegt. Auf diese Weise werden Fehler bei der Datenübertragung zwischen Mikroprozessor und Speichereinheit bzw. zwischen Speichereinheit und Schnittstelleneinheit erkannt.

[0041] Der CAN-Controller in dem Steuergerät kann als eigenständiger Baustein an den Mikroprozessor angeschlossen sein. Es ist aber auch möglich, daß der Mikroprozessor und der CAN-Controller in einem Baustein integriert sind. Derzeit sind eine Vielzahl von Mikroprozessoren bzw. Mikrocontrollern erhältlich, welche die CAN-Schnittstelle bzw. den CAN-Controller integriert haben. Der CAN-Controller befindet sich dann "on board".

Zeichnung

[0042] Die Erfindung wird anhand der beigefügten Zeichnung näher erläutert. Es zeigt:

[0043] Fig. 1 eine bevorzugte Ausführungsform des erfindungsgemäßen Steuergeräts in schematischer Darstellung;

[0044] Fig. 2 eine bevorzugte Ausführungsform des erfindungsgemäßen CAN-Controllers in schematischer Darstellung;

[0045] Fig. 3 eine bevorzugte Ausführungsform des erfindungsgemäßen Verfahrens bei der Bearbeitung von über einen CAN-Bus empfangenen Daten im Flußdiagramm;

[0046] Fig. 4 eine bevorzugte Ausführungsform des erfindungsgemäßen Verfahrens bei der Bearbeitung von über einen CAN-Bus zu sendenden Daten im Flußdiagramm.

[0047] In Fig. 1 ist ein Steuergerät insgesamt mit der Bezugsziffer **10** bezeichnet. Das Steuergerät **10** ist an einen CAN-Bus **11** angekoppelt. Pfeile **12** verdeutlichen, daß eine bidirektionale Kommunikation zwischen dem Steuergerät **10** und dem CAN-Bus **11** möglich ist.

[0048] Das Steuergerät weist einen CAN-Controller **13**, einen Mikroprozessor **14** und eine Speicherelement **15**, in diesem Fall ein RAM-Baustein, auf. Die Komponenten im Steuergerät **10** sind über eine Datenleitung **16** miteinander verbunden. Der CAN-Controller **13** dient als Schnittstelle zum CAN-Bus **11**. Über diesen werden Daten zum CAN Bus **11** gesendet bzw. von diesem empfangen.

[0049] Der CAN-Controller **13** liest zunächst über den CAN-Bus **11** übermittelte Daten ein. Anschließend werden in dem CAN-Controller **13** zu den empfangenen Daten Prüfbits erzeugt. Die Daten werden mit den Prüfbits von dem Mikroprozessor **14** eingelesen. So kann festgestellt werden, ob bei der Datenübertragung vom CAN-Controller **13** zum Mikroprozessor **14** ein Fehler vorliegt.

[0050] Fig. 2 zeigt eine bevorzugte Ausführungsform des erfindungsgemäßen CAN-Controllers in schematischer Darstellung. Zu erkennen ist der CAN-Controller, insgesamt mit der Bezugsziffer **20** bezeichnet. Außerdem sind ein Mikroprozessor **21** und ein CAN Bus **22** dargestellt. Der CAN-Controller **20** weist eine Schnittstelleneinheit **23**, eine Speichereinheit **24**, eine Busschnittstelle **25** und eine elektronische Einheit **26** auf.

[0051] Die Schnittstelleneinheit **23**, in diesem Fall ein sogenannter CAN-Protocol-Cernel, stellt die eigentliche Schnittstelle zum CAN Bus **22** dar. Hier werden Daten zum CAN-Bus **22** gesendet bzw. vom CAN-Bus **22** empfangen. Die Kommunikation innerhalb des CAN-Controllers und zwischen CAN-Controller **20** und Mikroprozessor **21** erfolgt über Datenleitungen **27**. Die bidirektionale Verbindung zwischen der Schnittstelleneinheit **23** und dem CAN-Bus **22** wird durch Pfeile **28** verdeutlicht.

[0052] In der Speichereinheit **24**, typischerweise ein sogenanntes Mailbox RAM, werden die empfangenen und die zu sendenden Daten abgelegt. Die einzelnen Komponenten im CAN-Controller **20** sind durch die Logik in der Busschnittstelle **25** verbunden.

[0053] Die Datenübertragung zwischen der Schnittstelleneinheit **23** und der Speichereinheit **24** wird durch die elektronische Einheit **26**, vorzugsweise eine Zustandsmaschine (Statemachine), gesteuert. Der Mikroprozessor **21** holt empfangene Daten aus der Speichereinheit **24** bzw. legt zu sendende Daten dort ab.

[0054] Da bekannt ist, welche Daten bzw. Nachrichten empfangen und gesendet werden, ist für jede Nachricht ein Bereich in der Speichereinheit **24** vorgesehen.

[0055] Fig. 3 zeigt das erfindungsgemäße Verfahren bei der Bearbeitung von über einen CAN-Bus empfangenen Daten im Flußdiagramm.

[0056] In einem ersten Schritt **30** werden Daten von der Schnittstelleneinheit **23** empfangen.

[0057] Die empfangenen Daten repräsentieren beispielsweise einen Steuerbefehl.

[0058] In einem darauffolgenden Schritt **31** werden von der Schnittstelleneinheit **23** entsprechende Prüfbits erzeugt. Die Daten werden zusammen mit den dazugehörigen Prüfbits in einem Schritt **32** in der Speichereinheit **24** abgelegt. Die Daten mit den dazugehörigen Prüfbits werden von dem Mikroprozessor **21** in einem Schritt **33** ausgelesen. Der Mikroprozessor **21** entscheidet, ob es sich um sicherheitsrelevante Daten handelt oder nicht. Dies erfolgt in einem Schritt **34**. Handelt es sich um sicherheitsrelevante Daten, erfolgt in einem weiteren Verfahrensschritt **35** die Überprüfung der Prüfbits. Liegen keine sicherheitsrelevanten Daten vor, können diese ohne Überprüfung in einem Schritt **36** weiter verwertet werden. Beispielsweise wird dann der Steuerbefehl ausgeführt.

[0059] Ergibt die Überprüfung im Schritt **35**, daß kein Fehler bei der Übertragung vorliegt, werden die Daten im Schritt **36** verwertet. Wird ein Fehler festgestellt, werden in einem Schritt **37** entsprechende Maßnahmen eingeleitet. Gegebenenfalls werden die Daten dann erneut angefordert.

[0060] Fig. 4 zeigt ein Verfahren bei der Bearbeitung von über einen CAN-Bus zu sendenden Daten im Flußdiagramm.

[0061] In einem ersten Schritt **40** werden zu sendende Daten von dem Mikroprozessor **21** zur Verfügung gestellt.

[0062] Diese Daten repräsentieren zum Beispiel eine Rückmeldung nach einer durchgeführten Aktion oder den Status des Steuergeräts.

[0063] In einem anschließenden Schritt **41** entscheidet der Mikroprozessor **21**, ob es sich dabei um sicherheitsrelevante Daten handelt oder nicht. Entsprechend werden Prüfbits generiert.

[0064] Nachfolgend werden die Daten in einem Schritt **42** mit den Prüfbits in der Speichereinheit **24** abgelegt. Das Auslesen des Speicherinhalts durch die Schnittstelleneinheit **23** erfolgt in einem Schritt **43**. In einem anschließenden Schritt **44** erkennt die Schnittstelleneinheit **23** anhand der Prüfbits, ob es sich um sicherheitsrelevante Daten handelt oder nicht. Sind die Daten sicherheitsrelevant, erfolgt in einem Schritt **45** die Überprüfung der Prüfbits. Sind die Daten nicht sicherheitsrelevant, werden diese ohne weitere Prüfung auf den CAN Bus **22** in einem Schritt **46** gegeben. Wird bei der Überprüfung der Prüfbits im Schritt **45** ein Fehler festgestellt, werden in einem Schritt **47** entsprechende Maßnahmen eingeleitet. Ansonsten werden die Daten im Schritt **46** auf den CAN Bus **22** gegeben.

[0065] Mit dem erfindungsgemäßen Verfahren kann die bei bekannten Verfahren bestehende Lücke in der Fehlerüberwachung geschlossen werden. Zur Erkennung von Fehlern bei der Datenübertragung innerhalb des CAN-Controllers **22** wird zumindest ein weiteres Prüfbit eingeführt. Die Prüfbits werden jeweils zusammen mit den Daten in der Speichereinheit **26** abgelegt. Beim Empfangen von Daten über den CAN Bus **22** werden die Prüfbits von der Schnittstelleneinheit **23** erzeugt und von dem Mikroprozessor **21** geprüft. Beim Senden werden die Prüfbits von dem Mikroprozessor **21** erzeugt und von der Schnittstelleneinheit **23** geprüft.

[0066] Um die besonders störanfällige Übertragung zwischen den CAN-Controllern **22**, die an den CAN-Bus **22** angekoppelt sind, zu überwachen, können selbstverständlich zusätzlich bekannte Verfahren angewandt werden. So überprüft typischerweise die Schnittstelleneinheit **23** die Prüfbits eingehender Daten und erzeugt Prüfbits beim Senden von Daten. Auf diese Weise werden Fehler bei der Datenübertragung über den CAN-Bus festgestellt.

[0067] Mit dem erfindungsgemäßen Verfahren kann also eine lückenlose Überwachung der Datenübertragung von Mikroprozessor zu Mikroprozessor sichergestellt werden. Die Sicherheit eines Systems kann dadurch weiter erhöht werden. Die Verbesserung läßt sich auch anhand einer FMEA-Analyse klar aufzeigen.

[0068] Eine FMEA-Analyse stellt ein Vorgehensmodell bei der Entwicklung von Systemen dar. Bei die-

ser Analyse werden bereits bei der Systementwicklung mögliche Fehler mit in die Überlegungen einbezogen. Die Analyse hilft somit, Systeme zu entwickeln, die wenig fehleranfällig sind. Außerdem ist es möglich, Systeme hinsichtlich ihrer Fehleranfälligkeit zu bewerten.

[0069] Es muß nach darauf hingewiesen werden, daß die auf dem CAN-Bus **22** verwendeten Prüfbits für eine interne Datenübertragung nicht genutzt werden, da hier neben den eigentlichen Daten auch die Werte des Übertragungsrahmens eingehen. Die Weiterleitung des Datenrahmens an den Mikroprozessor **21** wäre aber ein höherer Aufwand als die Neuberechnung der Prüfbits.

[0070] Ein besonderer Vorteil ist auch, daß zur effektiven Übertragung nicht-kritischer Daten das Verfahren abschaltbar ist. Werden bspw. als Prüfbits eine Prüfsumme berechnet, sieht eine einfache Implementierung folgendermaßen aus. Die Prüfsummenberechnung und Prüfung in der Schnittstelleneinheit **23** erfolgt in Hardware. Prüfsummen, die identisch Null sind, werden von der Hardware ignoriert.

[0071] Beim Senden von Daten entscheidet der Mikroprozessor **21**, ob eine sicherheitsrelevante Übertragung vorliegt und erzeugt entsprechend die Prüfbits. Liegen keine sicherheitsrelevanten Daten vor, wird von dem Mikroprozessor ohne Rechenaufwand eine Prüfsumme Null erzeugt. Prüfsummen, die identisch Null sind, werden dann in der Schnittstelleneinheit **23** ignoriert.

[0072] Beim Empfangen bleibt es dem Mikroprozessor **21** überlassen, ob die von der Hardware in der Schnittstelleneinheit **23** ermittelte Prüfsumme ausgewertet wird.

[0073] Das erfindungsgemäße Verfahren erlaubt somit eine permanente Überwachung des gesamten Übertragungsweges.

Patentansprüche

1. Verfahren zum Erkennen von Fehlern bei der Datenübertragung innerhalb eines CAN-Controllers (**13**, **20**), insbesondere bei der Datenübertragung zwischen einem Mikroprozessor (**14**, **21**) und einer Schnittstelleneinheit (**23**), wobei für über einen CAN-Bus (**11**, **22**) zu sendende Daten von dem Mikroprozessor (**14**, **21**) zumindest ein Prüfbit erzeugt wird, das zur Sicherstellung der Konsistenz der übertragenen Daten überprüft werden kann, **dadurch gekennzeichnet**, dass das erzeugte Prüfbit der Schnittstelleneinheit (**23**) anzeigt, ob sicherheitsrelevante Daten vorliegen, wobei die Schnittstelleneinheit (**23**) das mindestens eine Prüfbit bei sicherheitsrelevanten Daten überprüft.

2. Verfahren nach Anspruch 1, bei dem zunächst die über einen CAN-Bus (11, 22) zu sendenden Daten von dem Mikroprozessor (14, 21) zur Verfügung gestellt werden und die Daten mit dem mindestens einen Prüfbit in einer Speichereinheit (24) abgelegt werden, anschließend die Daten und das mindestens eine Prüfbit von einer Schnittstelleneinheit (23) ausgelesen werden und abschließend die Schnittstelleneinheit (23) das mindestens eine Prüfbit überprüft.

3. Verfahren nach Anspruch 1 oder 2, **dadurch gekennzeichnet**, dass die Schnittstelleneinheit (23) das mindestens eine Prüfbit bei nicht sicherheitsrelevanten Daten nicht überprüft.

4. Verfahren nach Anspruch 1, bei dem zunächst auf einem CAN-Bus (11, 22) befindliche Daten von der Schnittstelleneinheit (23) empfangen werden, anschließend zumindest ein weiteres Prüfbit erzeugt wird, die Daten mit dem mindestens einen weiteren Prüfbit in einer Speichereinheit (24) abgelegt werden, anschließend die Daten mit dem mindestens einen weiteren Prüfbit von dem Mikroprozessor (14, 21) eingelesen werden und schließlich vom Mikroprozessor (14, 21) das mindestens eine weitere Prüfbit überprüft werden kann.

5. Verfahren nach Anspruch 4, **dadurch gekennzeichnet**, dass der Mikroprozessor (14, 21) feststellt, ob die empfangenen Daten sicherheitsrelevant sind.

6. Verfahren nach Anspruch 5, **dadurch gekennzeichnet**, dass der Mikroprozessor (14, 21) bei sicherheitsrelevanten Daten das mindestens eine weitere Prüfbit überprüft.

7. Verfahren nach Anspruch 5 oder 6, **dadurch gekennzeichnet**, dass der Mikroprozessor (14, 21) bei nicht sicherheitsrelevanten Daten das mindestens eine weitere Prüfbit nicht überprüft.

8. CAN-Controller (13, 20) mit einer Schnittstelleneinheit (23) für den Datenaustausch mit einem CAN Bus (11, 22), einer Speichereinheit (24), in welcher empfangene Daten und zu sendende Daten abgelegt sind, und einer elektronischen Einheit (26) zum Steuern der Datenübertragung zwischen der Speichereinheit (24) und der Schnittstelleneinheit (23), **dadurch gekennzeichnet**, dass die Schnittstelleneinheit (23) Mittel zum Überprüfen von mindestens einem Prüfbit für zu sendende Daten aufweist, wobei das Prüfbit der Schnittstelleneinheit (23) anzeigt, ob sicherheitsrelevante Daten vorliegen, wobei die Schnittstelleneinheit (23) das mindestens eine Prüfbit bei sicherheitsrelevanten Daten überprüft.

9. CAN-Controller nach Anspruch 8, **dadurch gekennzeichnet**, dass die Schnittstelleneinheit (23)

Mittel zum Erzeugen von mindestens einem weiteren Prüfbit für empfangene Daten aufweist.

10. CAN-Controller (13, 20) nach Anspruch 8 oder 9, **dadurch gekennzeichnet**, dass die elektronische Einheit (26) eine Zustandsmaschine ist.

11. CAN-Controller (13, 20) nach einem der Ansprüche 8 bis 10, **dadurch gekennzeichnet**, dass eine Busschnittstelle (25) vorgesehen ist, deren interne Logik zur Datenübertragung zwischen der Schnittstelleneinheit (23), der Speichereinheit (24) der elektronischen Einheit (26) und einem Mikroprozessor (21) dient.

12. Steuergerät mit einem CAN-Controller (13, 20) nach einem der Ansprüche 8 bis 11, einem Mikroprozessor (14, 21) und einer Speichereinrichtung (15), **dadurch gekennzeichnet**, dass der Mikroprozessor (14, 21) Mittel zum Erzeugen und Überprüfen von Prüfbits aufweist.

13. Steuergerät nach Anspruch 12, **dadurch gekennzeichnet**, dass der Mikroprozessor (14, 21) und der CAN-Controller (13, 20) in einem Baustein integriert sind.

Es folgen 3 Seiten Zeichnungen

Anhängende Zeichnungen

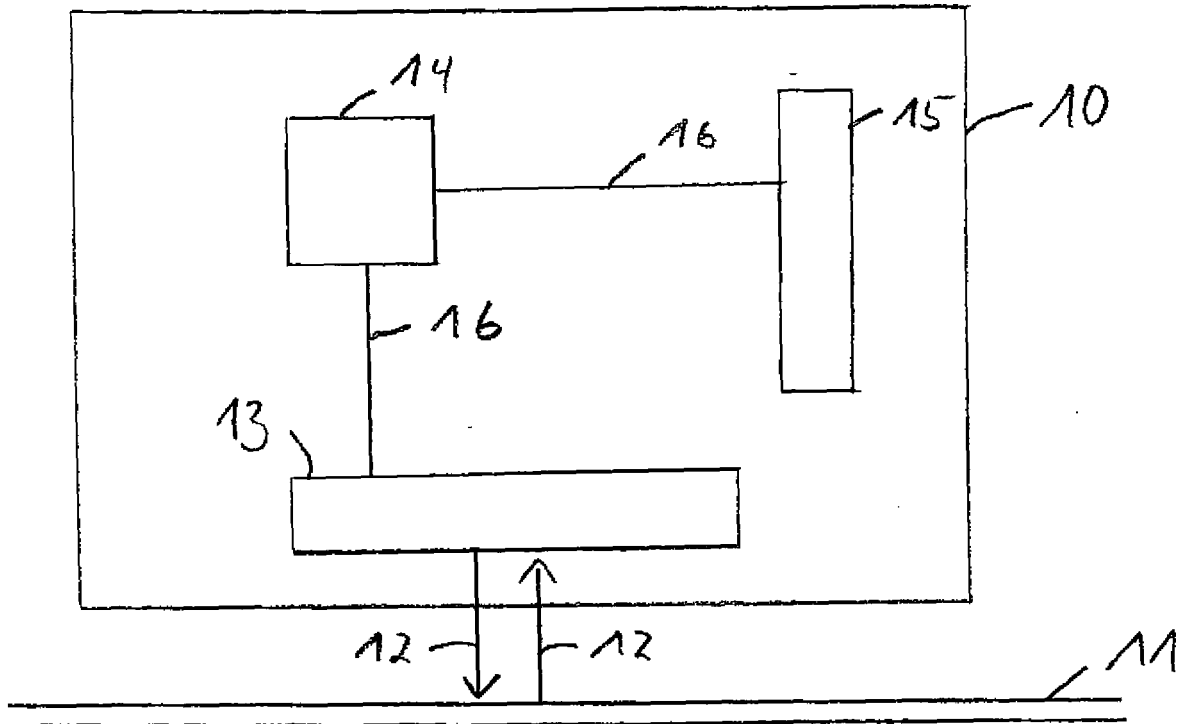


Fig. 1

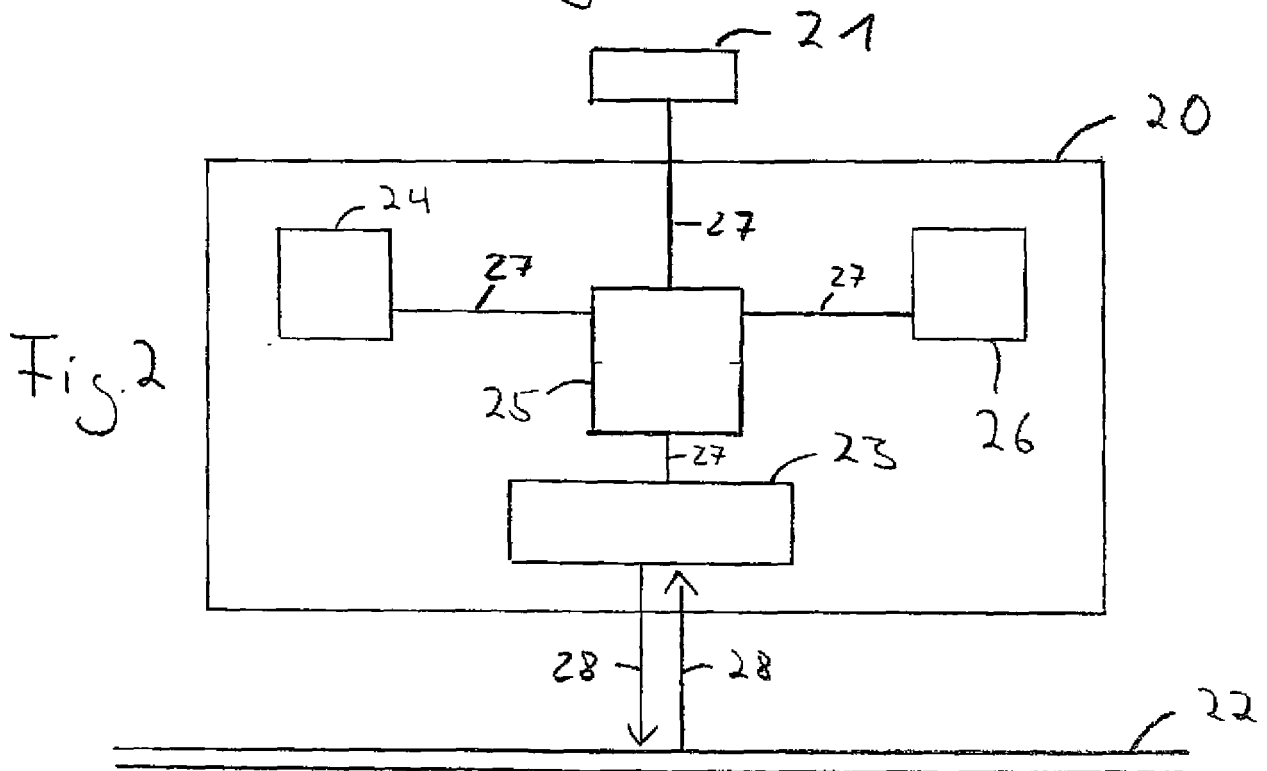


Fig. 2

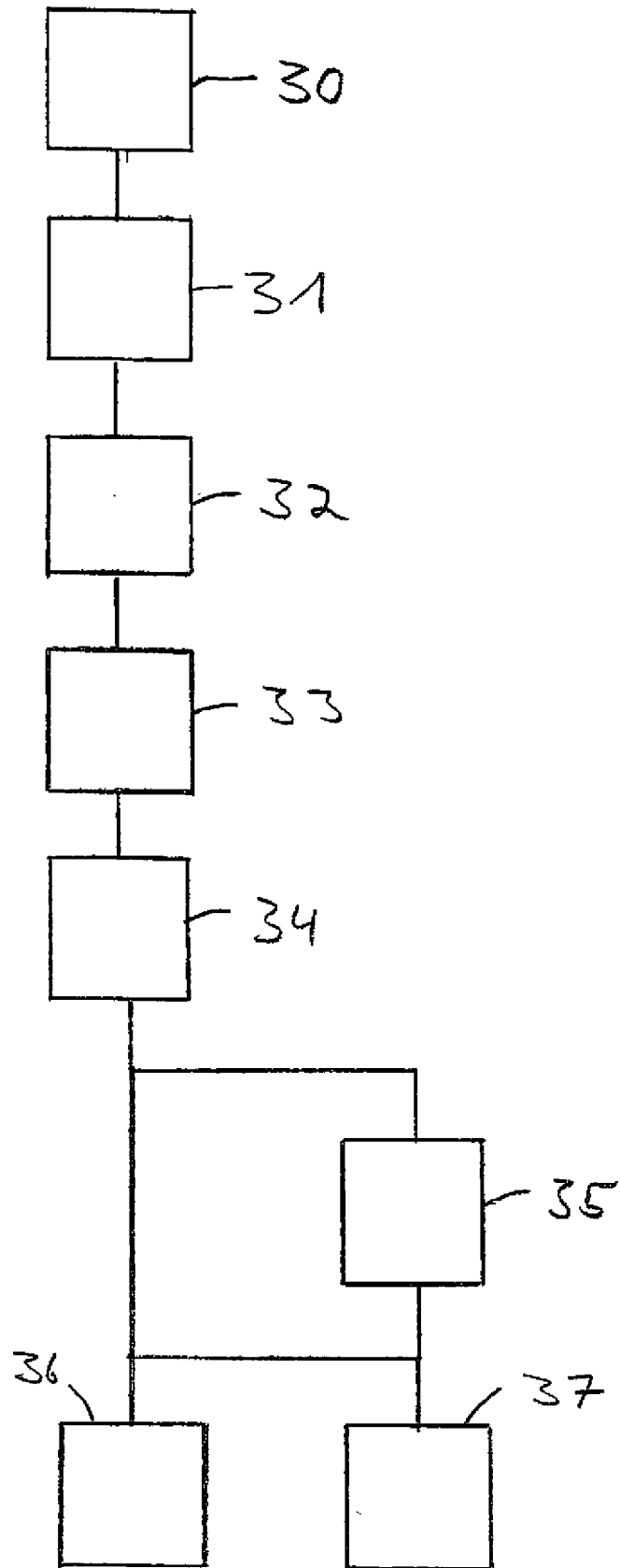


Fig. 3

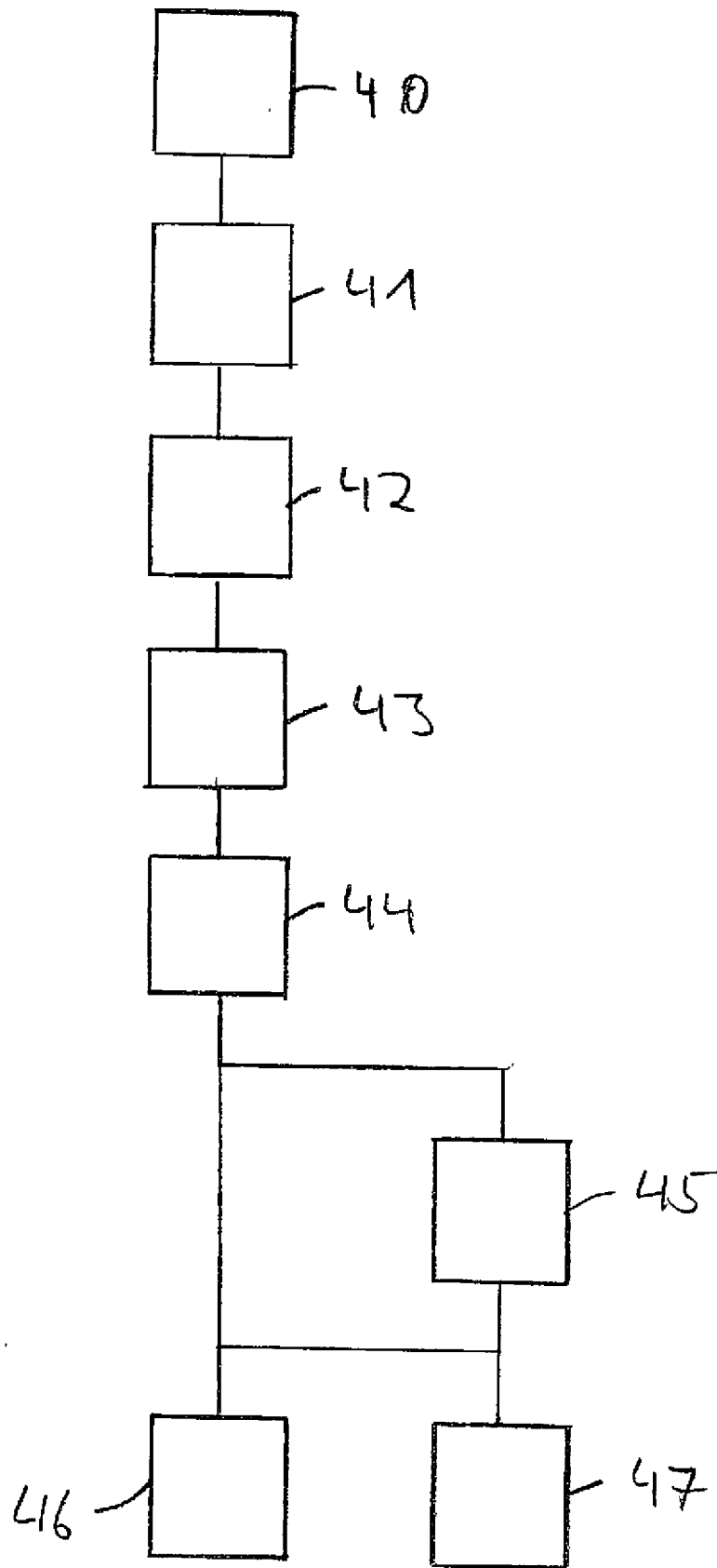


Fig. 4