



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 10 2006 039 461 A1** 2008.02.28

(12)

Offenlegungsschrift

(21) Aktenzeichen: **10 2006 039 461.5**

(22) Anmeldetag: **23.08.2006**

(43) Offenlegungstag: **28.02.2008**

(51) Int Cl.⁸: **G06F 9/445** (2006.01)

(71) Anmelder:

**Endress + Hauser GmbH + Co. KG, 79689
Maulburg, DE**

(72) Erfinder:

**Ferraro, Franco, 79739 Schwörstadt, DE;
Uppenkamp, Kaj, 79585 Steinen, DE; Rombach,
Walter, 79618 Rheinfelden, DE**

(74) Vertreter:

Andres, A., Pat.-Anw., 79576 Weil am Rhein

(56) Für die Beurteilung der Patentfähigkeit in Betracht zu ziehende Druckschriften:

DE 102 49 177 A1

DE 100 14 561 A1

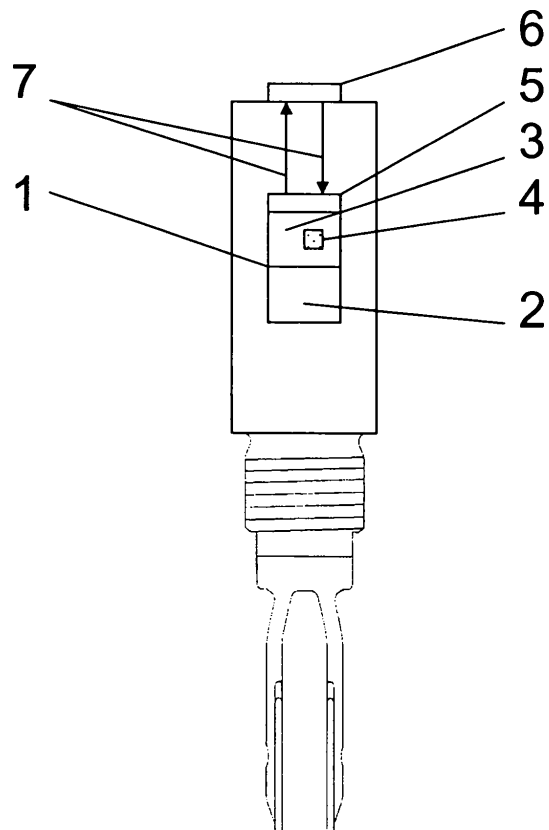
US 54 40 632

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Rechercheantrag gemäß § 43 Abs. 1 Satz 1 PatG ist gestellt.

(54) Bezeichnung: **Verfahren zur Programmierung eines Mikrocontrollers**

(57) Zusammenfassung: Die Erfindung bezieht sich auf ein Verfahren zur Programmierung eines Mikrocontrollers (1), wobei der Mikrocontroller (1) einen fest implementierten Bootstrap-Loader in einem Bootstrap-Loader-Speicherbereich (2) aufweist, wobei der Mikrocontroller (1) eine fest vorgegebene Initiierungsverbindung (10) aufweist, über welche das Ausführen des Bootstrap-Loaders aktivierbar ist, wobei der Mikrocontroller (1) eine fest vorgegebene Programmierverbindung (11) aufweist, über welche der Bootstrap-Loader vom Bootstrap-Loader zu verarbeitende Daten empfängt, und wobei der Mikrocontroller (1) einen Programm-Speicherbereich (3) für die Speicherung von Programmen aufweist. Die Erfindung beinhaltet, dass Daten, welche den Mikrocontroller (1) erreichen, von einem Aufruf-Programm (4) im Programm-Speicherbereich (3) ausgewertet werden, dass in dem Fall, dass als Daten den Mikrocontroller (1) ein Programmierbefehl erreicht, das Aufruf-Programm (4) den Bootstrap-Loader aktiviert, und dass der Bootstrap-Loader die Daten in den Programm-Speicherbereich (3) schreibt, die den Mikrocontroller (1) über die Programmierverbindung (11) erreichen. Weiterhin bezieht sich die Erfindung auf eine Vorrichtung zur Bestimmung und/oder Überwachung einer Prozessgröße.



Beschreibung

[0001] Die Erfindung bezieht sich auf ein Verfahren zur Programmierung eines Mikrocontrollers, wobei der Mikrocontroller einen fest implementierten Bootstrap-Loader in einem Bootstrap-Loader-Speicherbereich aufweist, wobei der Mikrocontroller mindestens eine fest vorgegebene Initiierungsverbindung aufweist, über welche das Ausführen des Bootstrap-Loader aktivierbar ist, wobei der Mikrocontroller mindestens eine fest vorgegebene Programmierverbindung aufweist, über welche der Bootstrap-Loader mindestens vom Bootstrap-Loader zu verarbeitende Daten empfängt, und wobei der Mikrocontroller einen Programm-Speicherbereich für die Speicherung von Programmen aufweist. Weiterhin bezieht sich die Erfindung auf eine Vorrichtung zur Bestimmung und/oder Überwachung mindestens einer Prozessgröße, mit mindestens einer Daten-Schnittstelle zur Ein/Ausgabe von Daten, und mit mindestens einem Mikrocontroller, wobei der Mikrocontroller einen Bootstrap-Loader-Speicherbereich mit einem fest implementierten Bootstrap-Loader aufweist, wobei der Mikrocontroller mindestens eine fest vorgegebene Initiierungsverbindung aufweist, über welche das Ausführen des Bootstrap-Loaders aktivierbar ist, wobei der Mikrocontroller mindestens eine fest vorgegebene Programmierverbindung aufweist, über welche der Bootstrap-Loader mindestens vom Bootstrap-Loader zu verarbeitende Daten empfängt, wobei der Mikrocontroller mindestens einen Programm-Speicherbereich für die Speicherung von Programmen aufweist, wobei der Mikrocontroller mindestens eine Ein/Ausgabeverbindung aufweist, welche zur Ein- und/oder Ausgabe von Daten dient, und wobei die Daten-Schnittstelle mit der Ein-/Ausgabeverbindung verbunden ist.

[0002] Im Stand der Technik ist eine Vielzahl von Mikrocontrollern bekannt, und auch in der modernen Prozess- und Automatisierungstechnik findet sich eine große Reihe von Messgeräten, welche mit einem Mikrocontroller ausgestattet sind. Eine besondere Thematik liegt in der Programmierung der Mikrocontroller oder im sog. Flashen. D.h. die vom Mikrocontroller auszuführenden Programme müssen in einem Speicherbereich abgelegt werden. Dies kann bei der Fertigung erforderlich sein oder auch bei einem Update der Programme beim bereits gefertigten und ggf. auch schon am Einsatzort installierten Gerät. Hierfür wird ein sog. Bootstrap-Loader (BSL) verwendet, welcher Programmcode in den Programm-Speicherbereich des Mikrocontrollers schreibt. Hierfür weist der Mikrocontroller eigens eine Programmierverbindung und eine Initiierungsverbindung auf, welche beide – oft exklusiv – dem BSL zugeordnet sind. Der BSL kann aktiv werden, wenn der Mikrocontroller beispielsweise nach einem Reset neu mit Spannung versorgt wird oder wenn an der Initiierungsverbindung ein Aktivierungssignal anliegt oder

eine Aktivierungssequenz angelegt wird. Erfolgt dieses Signal, so empfängt der BSL über die Programmierverbindung die Daten, Programme oder Programmteile, die entsprechend in den Programm-Speicherbereich zu schreiben sind.

[0003] Ein solcher BSL ist üblicherweise fest im Mikrocontroller vom Hersteller implementiert, d.h. fest vorgegeben. Für diesen implementierten BSL ist jedoch die Aktivierung über die Initiierungsverbindung erforderlich. Um diese Verbindung einzusparen, ist es im Stand der Technik bekannt, einen eigenen BSL zu verwenden, welcher Platz im Programm-Speicherbereich findet. Dieser eigene BSL lässt sich dann beispielsweise auch über die Kommunikationsverbindung des Mikrocontrollers aktivieren. Unter der Kommunikationsverbindung des Mikrocontrollers sei dabei die Datenverbindung verstanden, welche der Mikrocontroller zur Ausübung seiner Programme verwendet, um beispielsweise Messwerte zu empfangen, um Steuersignale oder verarbeitete Werte auszugeben. Diese Variante des Standes der Technik spart somit die Kontaktierung der Initiierungsverbindung ein, aber der zusätzliche BSL reduziert den zur Verfügung stehenden Speicherplatz im Programm-Speicherbereich.

[0004] Die Aufgabe der Erfindung besteht somit darin, ein Verfahren zum Flashen eines Mikrocontrollers vorzuschlagen, bei welchem möglichst nur ein geringer Speicherbedarf erforderlich ist und bei dem möglichst wenige Verbindungsleitungen zum Mikrocontroller hergestellt werden müssen. Weiterhin besteht eine Aufgabe der Erfindung darin, eine Messvorrichtung mit einem Mikrocontroller anzugeben, bei welchem der Programmspeicher optimal verwendet wird und bei welchem nur wenige Leitungen hergestellt werden.

[0005] Die Erfindung löst die Aufgabe durch ein Verfahren zur Programmierung eines Mikrocontrollers, wobei der Mikrocontroller einen fest implementierten Bootstrap-Loader in einem Bootstrap-Loader-Speicherbereich aufweist, wobei der Mikrocontroller mindestens eine fest vorgegebene Initiierungsverbindung aufweist, über welche das Ausführen des Bootstrap-Loader aktivierbar ist, wobei der Mikrocontroller mindestens eine fest vorgegebene Programmierverbindung aufweist, über welche der Bootstrap-Loader mindestens vom Bootstrap-Loader zu verarbeitende Daten empfängt, und wobei der Mikrocontroller einen Programm-Speicherbereich für die Speicherung von Programmen aufweist. Das erfindungsgemäße Verfahren sieht dabei vor, dass Daten, welche den Mikrocontroller erreichen, von einem Aufruf-Programm im Programm-Speicherbereich ausgewertet werden, dass in dem Fall, dass als Daten den Mikrocontroller ein Programmierbefehl erreicht, das Aufruf-Programm den Bootstrap-Loader aktiviert, und dass der Bootstrap-Loader die Daten in

den Programm-Speicherbereich schreibt, die den Mikrocontroller über die Programmierverbindung erreichen.

[0006] Die Erfindung besteht somit darin, dass der Mikrocontroller durch ein Programm überprüft, ob ein Aktivierungssignal den Mikrocontroller erreicht. Dieses Programm befindet sich dabei im Programm-Speicherbereich. Das Aktivierungssignal wird dabei nicht über die Initiierungsverbindung geschickt, sondern beispielsweise über die Programmierverbindung oder die Kommunikationsverbindung, d.h. die Kontaktierung der Initiierungsverbindung ist nicht erforderlich. D.h. der Mikrocontroller überprüft die Daten in Hinsicht auf den Aktivierungsbefehl, welche den Mikrocontroller auf Verbindungen unterschiedlich zur Initiierungsverbindung erreichen. Erfolgt ein solcher Aktivierungsbefehl, d.h. sind Daten oder Programme neu in den Programm-Speicherbereich zu schreiben, so springt das Programm den vom Hersteller des Mikrocontrollers fest implementierten BSL direkt (beispielsweise durch einen „jump“-Befehl) an, d.h. der BSL wird nicht durch das Signal auf der Initiierungsverbindung aktiviert, sondern das Programm im Programm-Speicherbereich startet oder aktiviert den BSL direkt. Dies bringt den Vorteil mit sich, dass kein eigenständiger BSL erforderlich ist, indem der bereits vorhandenen BSL genutzt werden kann. Die vom BSL in den Programm-Speicherbereich zu schreibenden Daten erreichen den BSL in einer alternativen Ausgestaltung über die normale Datenverbindung/Kommunikationsverbindung erreichen.

[0007] Eine Ausgestaltung des erfindungsgemäßen Verfahrens sieht vor, dass die Daten, welche den Mikrocontroller über die Programmierverbindung erreichen, von dem Aufruf-Programm darauf ausgewertet werden, ob den Mikrocontroller ein Programmierbefehl erreicht. In dieser Ausgestaltung horcht das Programm also, ob auf der Programmierverbindung, über welche den fest implementierten BSL auf die von ihm zu schreibenden Daten erreichen, der Aktivierungsbefehl ergeht.

[0008] Die Erfindung löst die Aufgabe weiterhin durch eine Vorrichtung zur Bestimmung und/oder Überwachung mindestens einer Prozessgröße, mit mindestens einer Daten-Schnittstelle zur Ein-/Ausgabe von Daten, und mit mindestens einem Mikrocontroller, wobei der Mikrocontroller einen Bootstrap-Loader-Speicherbereich mit einem fest implementierten Bootstrap-Loader aufweist, wobei der Mikrocontroller mindestens eine fest vorgegebene Initiierungsverbindung aufweist, über welche das Ausführen des Bootstrap-Loaders aktivierbar ist, wobei der Mikrocontroller mindestens eine fest vorgegebene Programmierverbindung aufweist, über welche der Bootstrap-Loader mindestens vom Bootstrap-Loader zu verarbeitende Daten empfängt, wobei der Mikrocontroller mindestens einen Programm-Speicherbe-

reich für die Speicherung von Programmen aufweist, wobei der Mikrocontroller mindestens eine Ein-/Ausgabeverbindung aufweist, welche zur Ein- und/oder Ausgabe von Daten dient, und wobei die Daten-Schnittstelle mit der Ein/Ausgabeverbindung verbunden ist. Die Erfindung sieht dabei vor, dass in dem Programm-Speicherbereich ein Aufruf-Programm bzw. Abfrage-Programm abgelegt ist, wobei das Aufruf-/Abfrage-Programm derartig ausgestaltet ist, dass das Aufruf-Programm die über die Ein-/Ausgabeverbindung eingehenden Daten dahingehend abfragt, ob ein Programmierungsbefehl oder Aktivierungsbefehl ergeht, und dass das Aufruf-Programm in dem Fall, dass ein Programmierungsbefehl ergeht, den Bootstrap-Loader aktiviert. Dieses Aufruf-Programm bzw. Abfrage- bzw. Aufruf-/Abfrage-Programm reagiert somit auf den Aufruf, welcher den fest implementierten BSL eigentlich über die Initiierungsverbindung erreichen würde, und aktiviert dann den BSL, welcher anschließend wiederum die Daten, die ihn über die Programmierverbindung erreichen, in den Programm-Speicherbereich schreibt. Wesentlich ist jedoch, dass der Aktivierungs- oder Programmierungsbefehl gerade auf einer Verbindung unterschiedlich zur Initiierungsverbindung übertragen wird.

[0009] Eine Ausgestaltung der erfindungsgemäßen Vorrichtung sieht vor, dass die Ein-/Ausgabeverbindung mindestens die fest vorgegebene Programmierverbindung und eine Kommunikationsverbindung umfasst. In diesem Fall werden von dem Mikrocontroller zwei Verbindungen weggeführt: eine für den Datenverkehr mit dem fest implementierten BSL und eine für die übliche und während des Arbeitens des Mikrocontrollers auftretende Datenkommunikation mit dem Mikrocontroller. Die Programmierverbindung ist speziell für den BSL vorgesehen und die Kommunikationsverbindung dient den Programmen im Programm-Speicherbereich. Beachtlich ist, dass erfindungsgemäß die Initiierungsverbindung nicht nach außen geführt ist.

[0010] Eine Ausgestaltung der erfindungsgemäßen Vorrichtung beinhaltet, dass die Ein-/Ausgabeverbindung nur die fest vorgegebene Programmierverbindung umfasst. In dieser Variante wird effektiv nur eine einzige Verbindung vom Mikrocontroller weggeführt. In einer Ausgestaltung wird dabei die Kommunikationsverbindung mit der Programmierverbindung verbunden, so dass also alle eingehenden und ausgehenden Daten über einen Knotenpunkt laufen.

[0011] Eine Ausgestaltung der erfindungsgemäßen Vorrichtung sieht vor, dass die Programmier-Schnittstelle über zwei Leitungen mit der Ein/Ausgabeverbindung des Mikrocontrollers verbunden ist. Es ist somit eine Eingangs- und eine Ausgangsleitung ($R \times D$ und $T \times D$) vorgesehen.

[0012] Eine Ausgestaltung der erfindungsgemäßen Vorrichtung beinhaltet, dass das Aufruf-Programm derartig in dem Programm-Speicherbereich abgelegt ist, dass das Aufruf-Programm unüberschreibbar ist. Erfolgt das Updaten des Programmspeichers nicht ordnungsgemäß, so kann es dazu kommen – da sich das Aufruf-Programm in eben diesem Programmspeicher befindet –, dass das Messgerät bzw. der Mikrocontroller auf die erfindungsgemäße Art nicht mehr updatefähig ist. Ursache dafür kann beispielsweise eine Unterbrechung der Kommunikation, eine Unterspannung oder keine Spannung sein. Dafür ist das Aufruf-Programm an einem fest zugeordneten Speicherplatz abgelegt und wird bei einem Updaten nicht mehr überschrieben. Somit ist bei einem missglückten Updateversuch ein nochmaliger Versuch immer noch möglich.

[0013] Eine Ausgestaltung der erfindungsgemäßen Vorrichtung sieht vor, dass das Aufruf-Programm derartig in dem Programm-Speicherbereich abgelegt ist, dass ein Überschreiben des Aufruf-Programms die Erfüllung mindestens eines vorgebbaren Kontrollkriteriums voraussetzt. Bei dem Kontrollkriterium handelt es sich beispielsweise um ein Passwort oder um eine sonstige Sicherung, welche nur ganz gezielt das Überschreiben zulässt. Das Aufruf-Programm ist somit durch das Kontrollkriterium vor einem allgemeinen Überschreiben geschützt, insbesondere wird das Überschreiben bei einem Update beispielsweise so verhindert. Dies ist somit eine Alternative zur vorhergehenden Ausgestaltung.

[0014] Die Erfindung wird anhand der nachfolgenden Zeichnungen näher erläutert.

[0015] Es zeigt:

[0016] [Fig. 1](#): eine schematische Darstellung eines erfindungsgemäßen Messgerätes,

[0017] [Fig. 2](#): eine erste Variante der Verbindungen zum Mikrocontroller, und

[0018] [Fig. 3](#): eine zweite Variante für die Verbindungen.

[0019] [Fig. 1](#) zeigt schematisch ein Messgerät, bei welchem es sich als ein Beispiel um eine sog. Schwinggabel handelt. Die mechanisch schwingfähige Einheit wird zu Schwingungen angeregt, und durch die Wechselwirkung zwischen den Gabelzinken und dem Medium lassen sich aus den Kenngrößen der Schwingungen Prozessgrößen wie Füllstand, Dichte oder Viskosität bestimmen bzw. lassen sich Änderungen davon beobachten. Das Messgerät weist einen Mikrocontroller 1 auf, in welchem die Programme zum Betrieb des Messgerätes, zur Auswertung der Messdaten usw. abgelegt und von welchem diese auch ausgeführt werden. Der Mikrocontroller 1

weist eine Ein/Ausgabeverbindung 5 auf, über welche er kommuniziert und über welche er Daten bzw. Programme erhält. Über zwei Leitungen (R×D und T×D) ist diese Verbindung 5 mit einer Daten-Schnittstelle 6 verbunden, über welche somit Daten vom Messgerät abgegriffen werden können bzw. über welche z.B. vom Bedienpersonal oder von einer übergeordneten Leitwarte aus das Messgerät bzw. der Mikrocontroller 1 programmiert bzw. geflashed werden kann. Bei dem Mikrocontroller handelt es sich beispielsweise um einen MSP 430 der Firma Texas Instruments.

[0020] Der Mikrocontroller 1 weist zumindest zwei Speicherbereiche 2, 3 auf. In einem Bootstrap-Loader-Speicherbereich 2 befindet sich der von der Herstellerfirma fest eingesetzte Bootstrap-Loader (BSL). Dieser BSL lässt sich entweder starten, nachdem der Mikrocontroller 1 neu mit Spannung versorgt wird oder durch einen besonderen Steuerbefehl, wobei der Mikrocontroller 1 bereits mit Spannung betrieben wird und bereits arbeitet. Diese Aktivierung erfolgt (siehe [Fig. 2](#)) durch eine speziell hierfür vorgesehene Initiierungsverbindung 10. Diese bilden im in [Fig. 2](#) gezeigten Fall zwei Leitungen oder Pins, auf welche ein spezielles Signal oder eine spezielle Sequenz gegeben wird, welches den BSL aktiviert, um Daten bzw. Programme von der wiederum speziell für diesen Zweck vorgesehenen Programmierverbindung 11 zu empfangen und diese in den Programm-Speicherbereich 3 zu schreiben. D.h. der fest implementierte BSL verfügt standardmäßig über zwei Verbindungen: eine für die Aktivierung, die andere für die Daten bzw. Programme. Weiter ist in der [Fig. 1](#) der zweite, freie Programm-Speicherbereich 3 dargestellt, in welchem die Programme und Daten liegen.

[0021] Im Stand der Technik bestehen zwei Möglichkeiten für die Programmierung eines Mikrocontrollers 1 bzw. für ein Überschreiben oder Flashen des Programmspeichers. Die eine Möglichkeit verwendet den fest implementierten BSL, wofür jedoch die Initiierungsverbindung 10 und die Programmierverbindung 11 kontaktiert bzw. aus dem den Mikrocontroller 1 umfassenden Messgerät herausgeführt werden müssen. Die andere Möglichkeit verwendet einen eigenen BSL, welcher im Programm-Speicherbereich 3 abgelegt wird und dabei unabhängig vom fest implementierten BSL ist. Diese zweite Möglichkeit macht somit keine Kontaktierung der beiden o.g. Verbindungen erforderlich. Nachteilig ist jedoch, dass der eigene BSL Speicherplatz benötigt.

[0022] Erfindungsgemäß ist ein Aufruf-Programm 4 im Programm-Speicherbereich 3 abgelegt. Dieses Programm 4 überwacht, ob den Mikrocontroller 1 über die Ein-/Ausgabeverbindung 5 ein Programmierbefehl erreicht und aktiviert ggf. den fest implementierten BSL. Die Idee ist somit, dass ohne die Kontaktierung der Initiierungsverbindung 10 der fest

implementierte BSL verwendet wird, indem er per Programm angesprochen und dadurch aktiviert wird. Damit ergeben sich die Vorteile, dass die Initiierungsverbindung **10** nicht kontaktiert werden muss und dass gleichzeitig eine Ersparnis an Speicherplatzbelegung erfolgt, da kein eigener BSL erforderlich ist. Die Speichergröße ist vom verwendeten Mitglied der Mikrocontroller-Familie, z.B. der MSP430-Familie abhängig und kann nach heutigem Stand der Technik zwischen 1kB und 120kB betragen. Ein zusätzlicher Bootloader weist beispielsweise ca. 8KB Programm-Code. Für die genannten Füllstandsmessgeräte sind Programmspeichergrößen zwischen 2kB und 32kB üblich.

[0023] Die [Fig. 2](#) und [Fig. 3](#) zeigen jeweils zwei Varianten für die Ausgestaltung der Ein/Ausgabeverbindung **5** des Mikrocontrollers **1**.

[0024] In der [Fig. 2](#) umfasst die Ein-/Ausgabeverbindung **5** sowohl die Programmierverbindung **11** des fest implementierten BSL, als auch die Kommunikationsverbindung **12**, über welche der Mikrocontroller **1** eigentlich kommuniziert und Daten austauscht. Bei der Kommunikationsverbindung **12** handelt es sich dabei beispielsweise um eine UART-Verbindung. Möglich ist auch eine I²C-, eine SPI- oder auch eine HART-Verbindung. In dieser Ausgestaltung wertet das Aufruf-Programm bzw. Abfrage- oder Aufruf/Abfrage-Programm **4** die Daten, die den Mikrocontroller **1** über die Programmierverbindung **11** oder über die Kommunikationsverbindung **12** erreichen, darauf aus, ob ein Programmier- bzw. Aktivierungsbefehl ergeht. In diesem Fall springt das Programm **4** den fest implementierten BSL direkt an, und veranlasst ihn damit, die Programme, die über die Programmierverbindung **11** eingehen, in den Programm-Speicherbereich **3** zu schreiben. Im Vergleich zur hier gezeigten erfindungsgemäßen Ausgestaltung besteht die Ein-/Ausgabeverbindung **5** im Stand der Technik unter Verwendung des fest implementierten BSL aus der Initiierungsverbindung **10**, der Programmierverbindung **11** und der Kommunikationsverbindung **12**, d.h. im Stand der Technik muss jeder dieser drei Verbindungen durch entsprechende Kontaktierung realisiert werden. In der Erfindung jedoch wird zumindest die Initiierungsverbindung **10** nicht kontaktiert, d.h. auch nicht über die Ein-/Ausgabeverbindung **5** herangeführt.

[0025] In der Variante der [Fig. 3](#) ist die Kommunikationsverbindung **12** mit der Programmierverbindung **11** direkt verbunden, d.h. die Ein/Ausgabeverbindung **5** weist nur zwei Leitungen auf. Ergeht nun in diesem Fall der Programmierbefehl, so springt das Aufruf-Programm **4** den fest implementierten BSL an und macht die Ein-/Ausgabeverbindung **5** frei für die Kommunikation mit dem vom Hersteller des Mikrocontrollers **1** fest implementierten BSL. In dieser Variante wird also nur eine Verbindung effektiv nach au-

ßen geführt, indem zwei der Verbindungen miteinander verbunden werden, die Initiierungsverbindung **10** wird jedoch ebenfalls wie in [Fig. 2](#) nicht kontaktiert. Die in [Fig. 2](#) und [Fig. 3](#) gezeigten Varianten können bei der Programmierung des Mikrocontrollers bei der Fertigung oder im fertigen Produkt für das Flashen verwendet werden.

Bezugszeichenliste

1	Mikrocontroller
2	Bootstrap-Loader-Speicherbereich
3	Programm-Speicherbereich
4	Aufruf-Programm
5	Ein-/Ausgabeverbindung
6	Daten-Schnittstelle
7	Leitung
10	Initiierungsverbindung
11	Programmierverbindung
12	Kommunikationsverbindung

Patentansprüche

1. Verfahren zur Programmierung eines Mikrocontrollers (**1**),
wobei der Mikrocontroller (**1**) einen fest implementierten Bootstrap-Loader in einem Bootstrap-Loader-Speicherbereich (**2**) aufweist,
wobei der Mikrocontroller (**1**) mindestens eine fest vorgegebene Initiierungsverbindung (**10**) aufweist, über welche das Ausführen des Bootstrap-Loader aktivierbar ist, wobei der Mikrocontroller (**1**) mindestens eine fest vorgegebene Programmierverbindung (**11**) aufweist, über welche der Bootstrap-Loader mindestens vom Bootstrap-Loader zu verarbeitende Daten empfängt, und
wobei der Mikrocontroller (**1**) einen Programm-Speicherbereich (**3**) für die Speicherung von Programmen aufweist,
dadurch gekennzeichnet,
dass Daten, welche den Mikrocontroller (**1**) erreichen, von einem Aufruf-Programm (**4**) im Programm-Speicherbereich (**3**) ausgewertet werden, dass in dem Fall, dass als Daten den Mikrocontroller (**1**) ein Programmierbefehl erreicht, das Aufruf-Programm (**4**) den Bootstrap-Loader aktiviert, und
dass der Bootstrap-Loader die Daten in den Programm-Speicherbereich (**3**) schreibt, die den Mikrocontroller (**1**) über die Programmierverbindung (**11**) erreichen.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Daten, welche den Mikrocontroller (**1**) über die Programmierverbindung (**11**) erreichen, von dem Aufruf-Programm (**4**) darauf ausgewertet werden, ob den Mikrocontroller (**1**) ein Programmierbefehl erreicht.

3. Vorrichtung zur Bestimmung und/oder Überwachung mindestens einer Prozessgröße, mit mindestens einer Daten-Schnittstelle (6) zur Ein-/Ausgabe von Daten, und mit mindestens einem Mikrocontroller (1), wobei der Mikrocontroller (1) einen Bootstrap-Loader-Speicherbereich (2) mit einem fest implementierten Bootstrap-Loader aufweist, wobei der Mikrocontroller (1) mindestens eine fest vorgegebene Initiierungsverbindung (10) aufweist, über welche das Ausführen des Bootstrap-Loaders aktivierbar ist, wobei der Mikrocontroller (1) mindestens eine fest vorgegebene Programmierverbindung (11) aufweist, über welche der Bootstrap-Loader mindestens vom Bootstrap-Loader zu verarbeitende Daten empfängt, wobei der Mikrocontroller (1) mindestens einen Programm-Speicherbereich (3) für die Speicherung von Programmen aufweist, wobei der Mikrocontroller (1) mindestens eine Ein-/Ausgabeverbindung (5) aufweist, welche zur Ein- und/oder Ausgabe von Daten dient, und wobei die Daten-Schnittstelle (6) mit der Ein-/Ausgabeverbindung (5) verbunden ist, dadurch gekennzeichnet, dass in dem Programm-Speicherbereich (3) ein Aufruf-Programm (4) abgelegt ist, wobei das Aufruf-Programm (4) derartig ausgestaltet ist, dass das Aufruf-Programm (4) die über die Ein-/Ausgabeverbindung (5) eingehenden Daten dahingehend abfragt, ob ein Programmierungsbefehl ergeht, und dass das Aufruf-Programm (4) in dem Fall, dass ein Programmierungsbefehl ergeht, den Bootstrap-Loader aktiviert.

4. Vorrichtung nach Anspruch 3, dadurch gekennzeichnet, dass die Ein-/Ausgabeverbindung (5) mindestens die fest vorgegebene Programmierverbindung (11) und eine Kommunikationsverbindung (12) umfasst.

5. Vorrichtung nach Anspruch 3, dadurch gekennzeichnet, dass die Ein-/Ausgabeverbindung (5) nur die fest vorgegebene Programmierverbindung (11) umfasst.

6. Vorrichtung nach Anspruch 3, dadurch gekennzeichnet, dass die Programmier-Schnittstelle (6) über zwei Leitungen (7) mit der Ein-/Ausgabeverbindung (5) des Mikrocontrollers (1) verbunden ist.

7. Vorrichtung nach mindestens einem der Ansprüche 3 bis 6, dadurch gekennzeichnet, dass das Aufruf-Programm (4) derartig in dem Programm-Speicherbereich (3) abgelegt ist, dass das Aufruf-Programm (4) unüberschreibbar ist.

8. Vorrichtung nach mindestens einem der An-

sprüche 3 bis 6, dadurch gekennzeichnet, dass das Aufruf-Programm (4) derartig in dem Programm-Speicherbereich (3) abgelegt ist, dass ein Überschreiben des Aufruf-Programms (4) die Erfüllung mindestens eines vorgebbaren Kontrollkriteriums voraussetzt.

Es folgen 3 Blatt Zeichnungen

Fig. 1

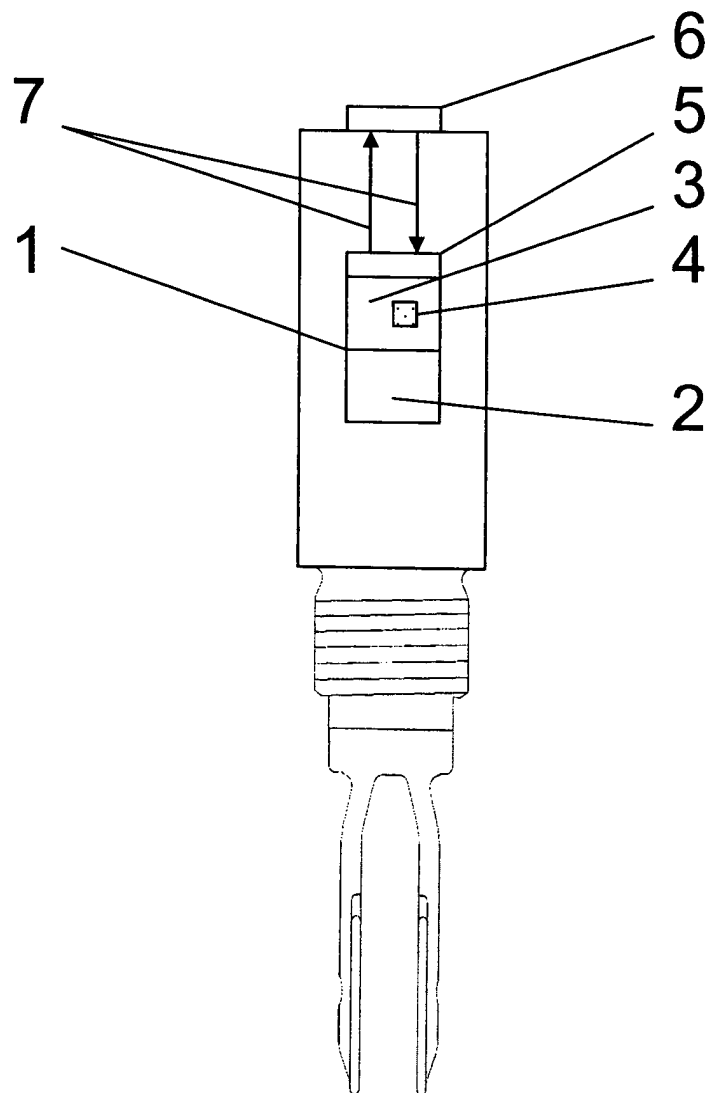


Fig. 2

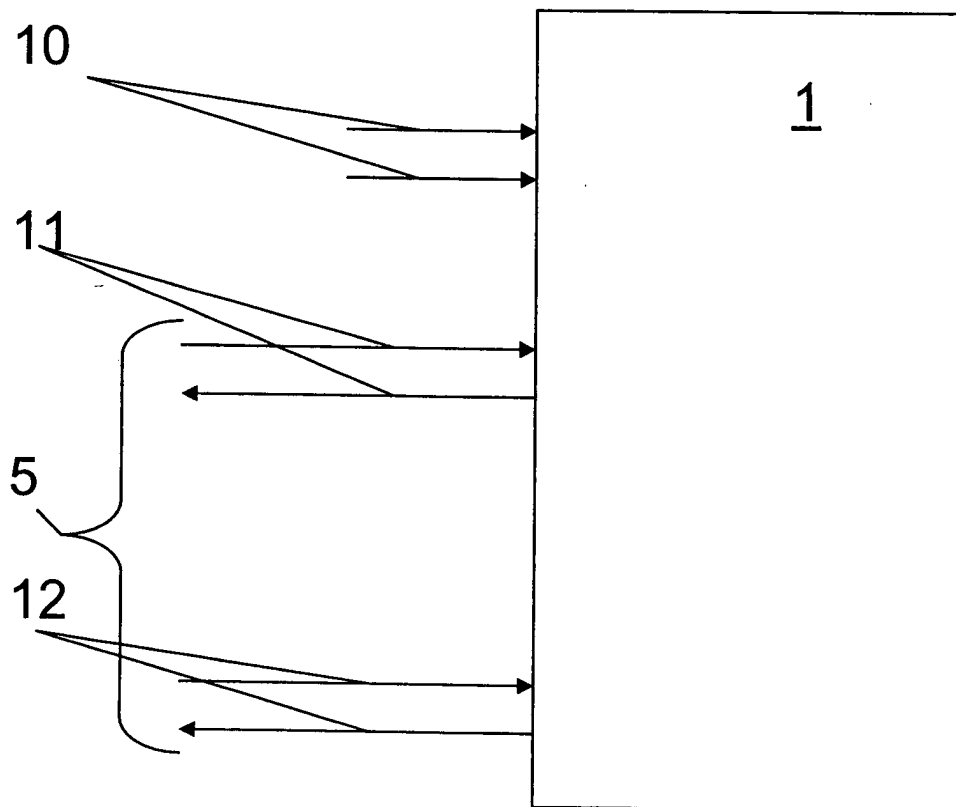


Fig. 3

