Wenn verfälschte Bits gefährlich werden : oder : wie man Daten in gestörter Umgebung zuverlässig speichert

Autor(en): **Volpe, Francesco P.**

Objekttyp: **Article**

Zeitschrift: **Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association Suisse des Electriciens, de l'Association des Entreprises électriques suisses**

Band (Jahr): **89 (1998)**

Heft 9

PDF erstellt am: **08.08.2024**

Persistenter Link: https://doi.org/10.5169/seals-902070

Bei seriellen, nichtflüchtigen Speichern kann es unter gestörten Betriebsbedingungen vorkommen, dass gespeicherte Daten unbeabsichtigt geändert werden. In der Automobilelektronik, in der diese Speicher teilweise sicherheitsrelevante Funktionen ausüben wie zum Beispiel im ABS (Anti-Lock Breaking System), im Airbag sowie in der Motor- und Getriebesteuerung kann eine Fehlprogrammierung aber nicht toleriert werden. Um solche Fehlprogrammierungen zu verhindern, verfügen serielle EEPROM von Siemens über einen Page Protection Mode [1], dessen Funktionsweise im vorliegenden Beitrag beschrieben wird.

Wenn verfälschte Bits gefährlich werden

Oder: Wie man Daten in gestörter Umgebung zuverlässig speichert

■ Francesco P. Volpe

Datenübertragung nach I2C-Bus-Protokoll

Serielle EEPROM mit Page Protection Mode sorgen für Datensicherheit im Automotive-Bereich. Die dafür notwendigen I2C-Bus-Protokolle (von Philips eingeführter Standard) lassen sich softwaremässig implementieren. Das verwendete Übertragungsprotokoll ist vollständig kompatibel zum I2C-Bus-Protokoll. Die Steuerung der Schutzbits erfolgt über einen leicht erweiterten Page-Write-Befehl. Beim normalen Page-Write-Befehl werden nach dem Startkriterium das Chip-Select-Byte zum Schreiben (CSW), dann die Wortadresse (WA) und anschliessend die Datenbytes gesendet (Bild 1a). Die Anzahl der maximal übertragenen Datenbytes hängt von der Page-Grösse des EEPROM ab. Der Empfang der einzelnen Bytes wird vom EEPROM durch Acknowledge(ACK)-Bits quittiert. Abgeschlossen wird die Übertragung durch eine vom Master gesendete Stoppbedingung.

Das Lesen, Schreiben und Löschen der Schutzbits geschieht entsprechend der Sequenz nach Bild 1b. Dabei wird die Sequenz für die Schutzbitfunktionen im Vergleich zum Page-Write-Befehl zwischen der Wortadresse und dem ersten Datenbyte um eine Startbedingung, ein CSW-Byte und ein neues Steuerbyte

Adresse des Autors
Dr.-Ing. ***Francesco P. Volpe***, Siemens AG
Geschäftszweig Hochfrequenztechnik
Balanstrasse 73, D-81539 München
Email francesco.volpe@siemens.de

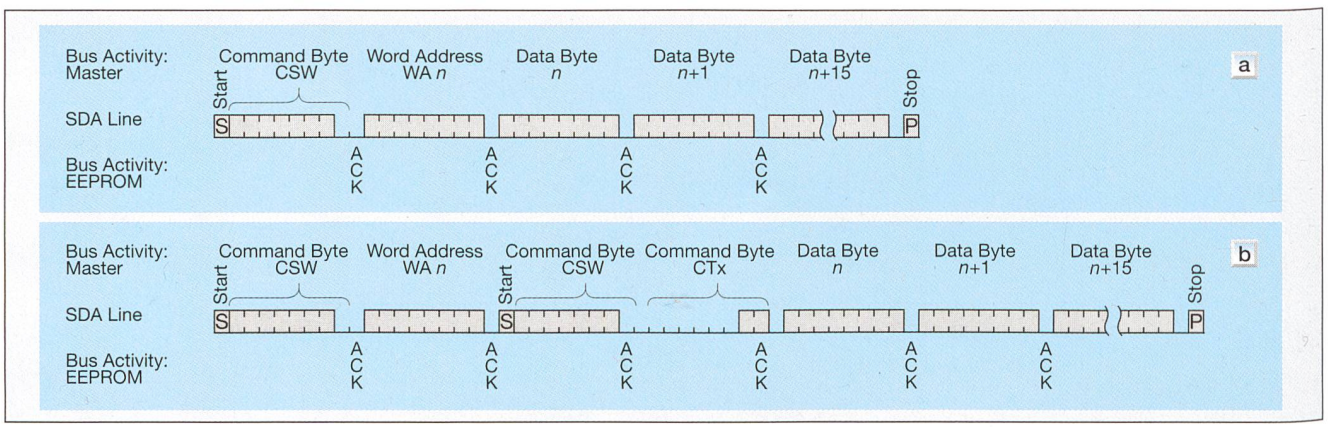

Bild 1 Zur Steuerung der Schutzbits wird der Page-Write-Befehl (a) leicht erweitert (b). Das Protokoll ist vollständig kompatibel zum I2C-Bus-Protokoll. In diesem Beispiel wurde eine Page-Grösse von 16 Byte angenommen.

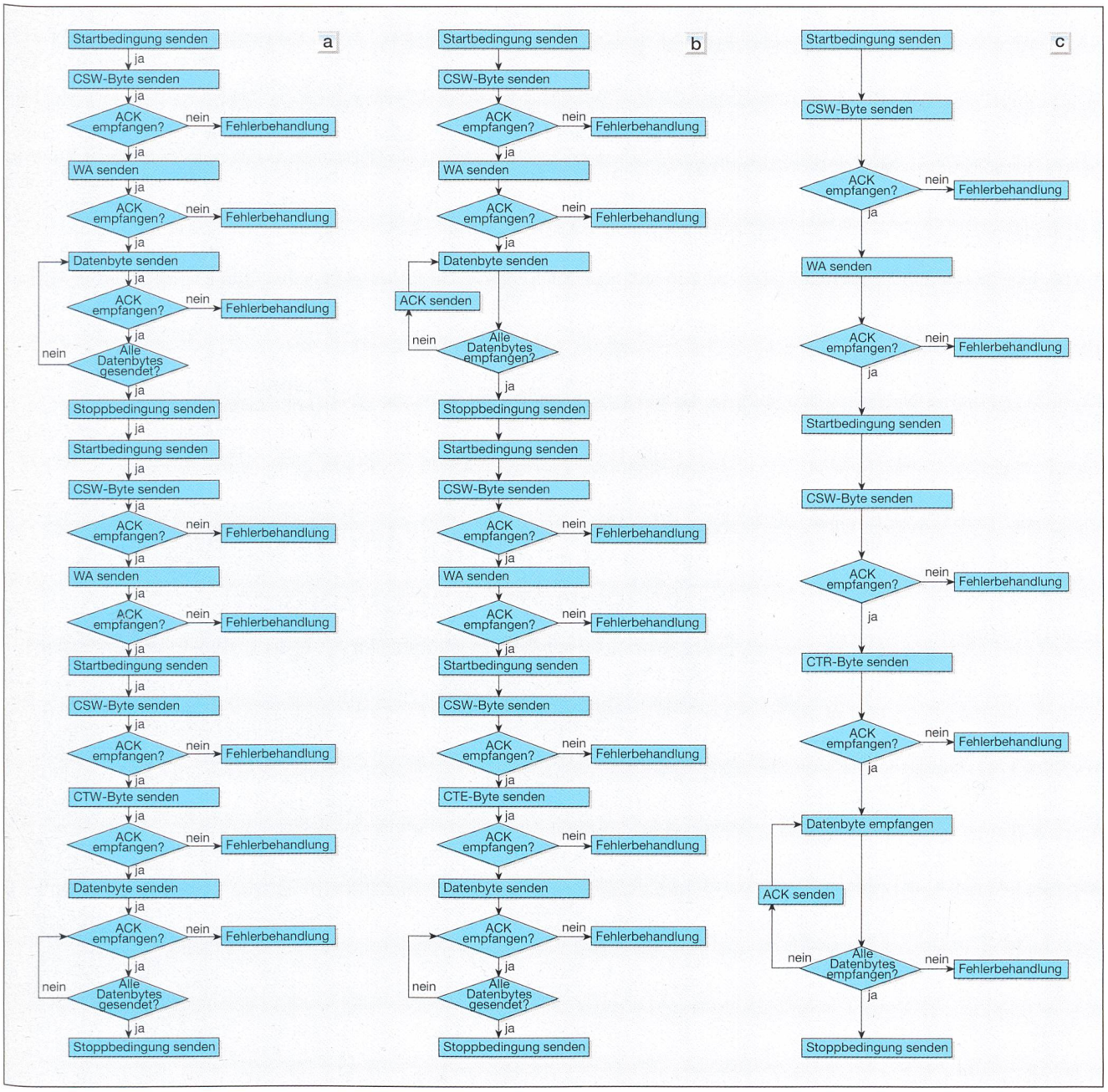

Bild 2 Um das Schutzbit zu setzen (a), zu löschen (b) oder zu lesen (c), wird jedes Mal der Inhalt der gesamten Page eingelesen.

(CTx) erweitert. So ist eine klare Unterscheidung zu I2C-Bus-Befehlen möglich. Die eigentliche Schutzbitfunktion ist in den untersten zwei Bits (B0 und B1) von CTx codiert. Mit Hilfe dieser zwei Bits lassen sich die Befehle zum Lesen (CTR), Schreiben (CTW) und Löschen (CTE) unterscheiden. Die restlichen sechs Bits können beliebig sein, da diese nicht decodiert werden (Tabelle I).

Schutzbit setzen

Soll zum Beispiel eine Page geschützt werden, muss das dieser Page zugeordnete Schutzbit gesetzt werden. Dazu werden zuerst die Daten in die zu schützende Page geschrieben (Bild 2a). Das kann entweder durch die Übertragung einzelner Bytes oder der gesamten Page auf einmal geschehen. Dabei wird angenommen, dass das Schutzbit dieser Page nicht gesetzt ist. In einem zweiten Schritt wird jetzt die Sequenz zum Setzen eines Schutzbits gesendet. Als Adresse im CSW- und WA-Byte muss die erste Stelle der Page übertragen werden. Das CTW-Byte ist entsprechend Tabelle I zu wählen. Anschliessend wird jedes Byte der Page erneut übertragen. Nach jeder Eingabe eines Datenbytes überprüft das EEPROM mit einem internen 8-Bit-Komparator die Übereinstimmung des gespeicherten mit dem erneut übertragenen Byte. Nur wenn alle acht Bits identisch sind, sendet das EEPROM ein ACK-Bit. Sind alle erneut übertragenen Daten mit den gespeicherten Daten identisch, wird das Schutzbit der Page gesetzt. Mit dieser Prozedur ist sichergestellt, dass einerseits bei der Übertragung der Daten kein Fehler aufgetreten ist und dass andererseits nur die gewünschten Daten geschützt werden. Es ist sehr unwahrscheinlich, dass zufällig 128 Bits (bei einer Page-Länge von 16 Byte) richtig übertragen werden oder dass 128 Bits gerade «richtig» gestört sind. Eine so geschützte Page lässt sich nur überschreiben, wenn das zugehörige Schutzbit wieder gelöscht wird.

CTx	B7	B6	B5	B4	B3	B2	B1	B0	Funktion
CTR	x	x	x	x	x	x	0	0	Schutzbit lesen
CTW	x	x	x	x	x	x	0	1	Schutzbit schreiben
CTE	x	x	x	x	x	x	1	1	Schutzbit löschen

Tabelle I Schutzbitfunktionen und deren Codierung im CTx-Byte

Schutzbit löschen

Zum Löschen eines Schutzbits müssen die gespeicherten Daten in der Page bekannt sein. Dies bedeutet, dass der Mikrocontroller unter Umständen den Inhalt der Page zunächst einmal lesen muss (Bild 2b). Anschliessend wird die Sequenz nach Bild 1b mit dem CTE-Byte nach Tabelle I gesendet. Nach jeder Eingabe eines Datenbytes überprüft das EEPROM, wie beschrieben, intern die Daten und sendet bei Übereinstimmung ein ACK-Bit. Bei Gleichheit aller Bytes der Page wird das Schutzbit gelöscht.

Schutzbit lesen

Der Status der Schutzbits lässt sich über den Lesebefehl CTR abfragen. Die Sequenz lehnt sich wieder an die in Bild 1b gezeigte an. Der Mikrocontroller muss in diesem Fall keine Daten an den Speicher senden, sondern erhält Daten von diesem (Bild 2c). Das höchstwertige Bit der 8-Bit-Pakete gibt den Zustand des Schutzbits der adressierten Page an. Nach dem Empfang muss der Mikrocontroller den Empfang mit einem ACK-Bit bestätigen. Der Speicher sendet so lange Daten, bis der Mikrocontroller diese nicht mehr bestätigt und die Datenübertragung mit einer Stoppbedingung beendet.

Die Überlegenheit des hier geschilderten Schutzkonzeptes gegenüber den konventionellen Schutzmassnahmen beruht unter anderem darauf, dass der Mikrocontroller keinen einfachen Programmbefehl zum Ein- und Ausschalten des Schutzes kennt. Es muss in jedem Fall eine zweiseitige Datenübertragung zwischen Mikrocontroller und EEPROM durchgeführt werden, und zwar beim Schreiben und beim Löschen des Schutzbits. Sollen zum Beispiel nicht die gesamten Bytes einer Page programmiert werden, muss man dennoch alle Daten der Page kennen, bevor eine Schutzfunktionsoperation ausgeführt werden kann. Die Tatsache, dass zum Ändern eines Schutzbits immer ein 128 Bits langer «Schlüssel» gesendet werden muss, erhöht die Datensicherheit enorm.

Literatur

[1] *Francesco P. Volpe, Jürgen Kuttruff, Hartmut Schrenk:* Datensicherheit bei seriellen EEPROMs: Page-Protection-Mode schützt zuverlässig, Components, Heft 5, 1997, S. 166–169.

Quand les données faussées deviennent dangereuses

Ou: comment stocker les données en toute sécurité en environnement perturbé

Avec les mémoires sérielles non volatiles, il peut arriver, dans des conditions perturbées, que des données stockées soient modifiées accidentellement. En électronique automobile, où ces mémoires ont des fonctions en partie vitales pour la sécurité, comme dans les systèmes de freinage ABS, les airbags ou la commande de moteur et de boîte automatique, les erreurs de programmation ne sauraient être tolérées. Afin d'éviter de telles erreurs, les EEPROM sériels de Siemens sont dotés d'un Page Protection Mode [1] dont le fonctionnement est décrit au présent article.