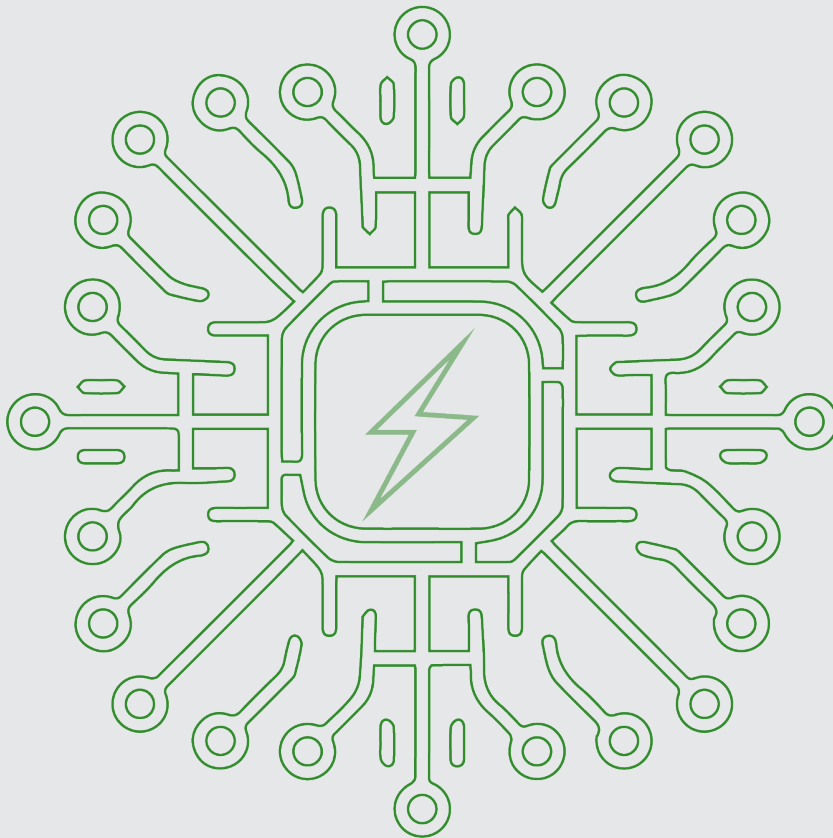


hello, whitepaper

Flash-Programmierung: Effizienz, High-Speed und Leistung für die Elektronik



Die Flash-Programmierung ist die verbreitetste Methode für das Programmieren einzelner PCBs, Mehrfachnutzen oder gehauster Baugruppen. Alle Details lesen Sie hier!

Inhalt

3

Relevanz

4

Aufbau eines Mikrocontrollers (MCU)

5

Flash-Speicher: Funktionsweise

6

Allgemeine Beschreibung der Flash-Programmierung

7

Flash-Programmierung im technischen Detail

8

In-System Programmierung (ISP)

10

Das Herzstück der Programmierung: Der In-System-Programmer

11

Die Kontrolle über den Flash-Prozess: Die Produktionssoftware

12

Margin-Verify von ProMik entwickelt

13

ProMik: Experte für die Flash-Programmierung

14

Use-Case: High-Speed Programmierung eines Clusters

15

Erfahren Sie mehr über Flash-Programmierung





Relevanz

Die Flash-Programmierung hat mit der zunehmenden Komplexität der Technologien eine immer größere Bedeutung erlangt. So hat sie sich von ihrer früheren Rolle im Rahmen des In-Circuit-Tests (ICT) zu einem eigenständigen Produktionsschritt entwickelt. Ihre Vorzüge, die sie von anderen nichtflüchtigen Programmiermethoden unterscheiden, sind die Möglichkeit zur Reprogrammierung, der schnelle Lesezugriff und die außergewöhnliche Datendichte.

Die Flash-Programmierung wird in fast allen Branchen eingesetzt. Die In-System-Programmierung, die die Programmierung nach der Platinenbestückung beschreibt, eignet sich besonders in Bereichen, in denen hohe Stückzahlen und komplexe Anwendungen gefordert sind. Ein ideales Beispiel hierfür ist die Automobilbranche.

Da es sich bei der Flash-Programmierung um einen hochtechnischen Prozess handelt, ist entsprechendes Know-how erforderlich. Die Zusammenarbeit mit Experten wie ProMik gewährleistet, dass Kunden in diesem Bereich bestens gerüstet sind.

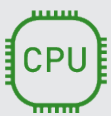
Aufbau eines Mikrcontrollers (MCU)

Ein Mikrocontroller (MCU) ist ein Halbleiterchip, der sowohl einen Prozessor als auch Peripherien beinhaltet. Seine Aufgabe ist das Messen, Steuern und Regeln von Applikationen.

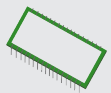
Die Funktionsweise ist wie folgt: Das Bauteil erkennt externe Signale über Kommunikationsprotokolle und reagiert darauf.

Man unterscheidet bei MCUs zwischen verschiedenen **Kernarten** (8-Bit, 16-Bit und 32-Bit) sowie **Architekturen** (z.B.: „Reduced Instruction Set Computer“ (RISC) oder „Acorn RISC Machine“(ARM)).

Mikrocontroller besitzen zudem unterschiedliche Bausteine. Dazu gehört die Central-Processing-Unit (CPU), verschiedene Speicher, I/O-Anschlüsse und Peripherien.



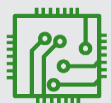
CPU: Zentraler Prozessor



Speicher: Mindestens ein RAM- und ein ROM- oder **Flash-Speicher**



I/O-Anschlüsse



Peripherien: z.B. **Kommunikations-** oder **Debug-Schnittstellen**

Flash-Speicher: Funktionsweise

Flash-Speicher sind nichtflüchtige Speicher, die sich von ROM-Speichern unterscheiden, da sie **reprogrammierbar** sind: Durch die Verwendung hoher Spannungen, können Daten gelöscht und erneut geschrieben werden. Flash-Speicher sind dadurch um einiges **flexibler**, haben **hohe Datenübertragungsraten** und **schnellere Reaktionszeiten** als ROM-Speicher.

Ein Flash-Speicher besteht aus verschiedenen Elementen – darunter ein Control- und ein Floating-Gate. Das Floating-Gate speichert die Informationen und wird von dem Rest des MCUs durch eine Oxidschicht getrennt. Außerdem enthält der Speicher eine Quelle, an der Ladungen ankommen und einen Abfluss, an welchem diese wieder abfließen. Details zu den Funktionen der Komponenten lesen Sie im [technischen Detail](#).

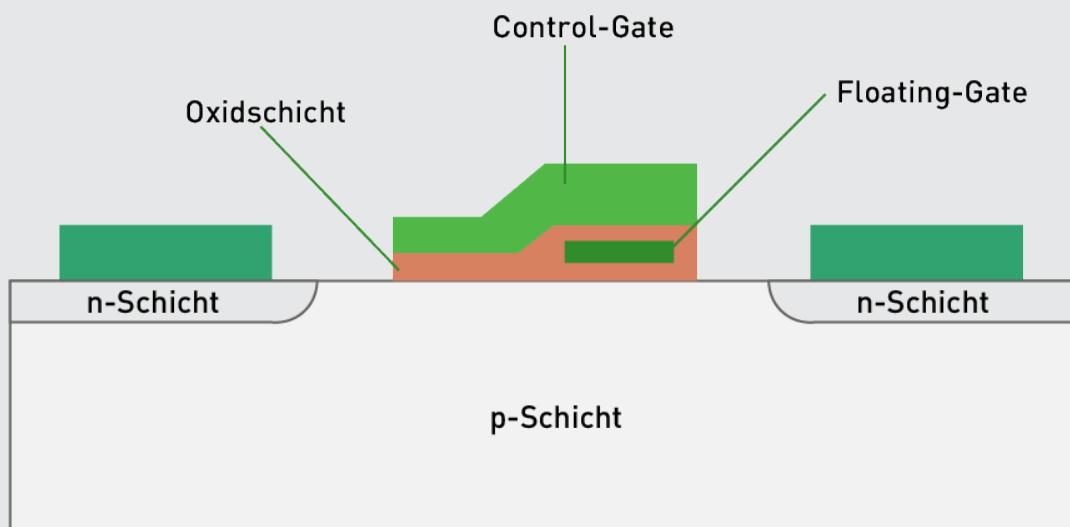


Abbildung: Flash-Speicher-Struktur

Es wird bei Flash-Speichern zwischen Not-AND-(**NAND-**) und Not-OR-(**NOR-**)-Speichern unterschieden.

Bei NAND-Flash-Speichern sind die Speicherzellen seriell geschaltet. Sie werden als Massenspeicher verwendet und reduzieren den Platzbedarf.

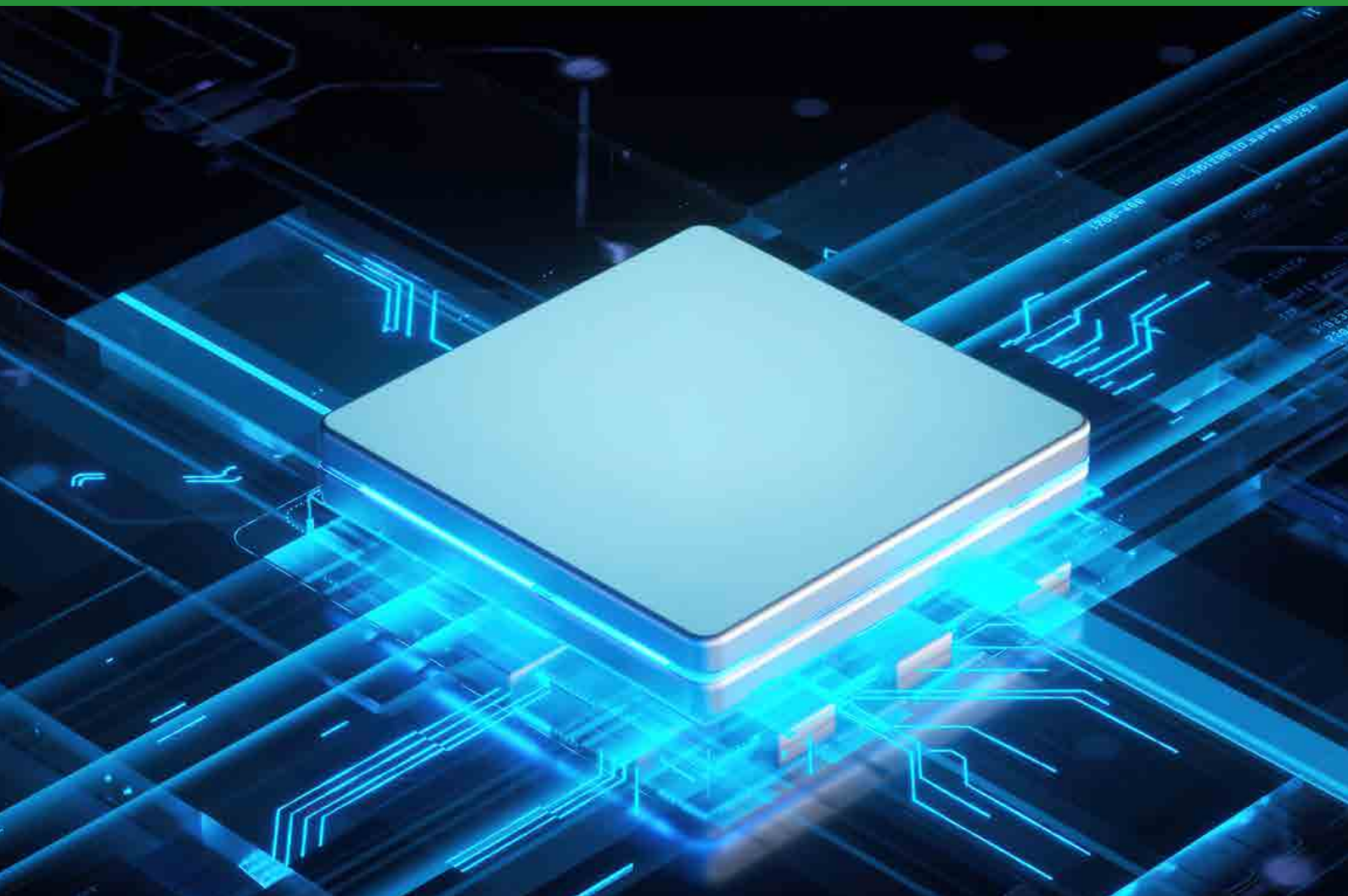
Die Speicherzellen von NOR-Flash-Speichern sind parallel geschaltet. Das ermöglicht einen direkten Zugang auf Daten, allerdings ist mehr Platz erforderlich.

Allgemeine Beschreibung der Flash-Programmierung

Zunächst wird die Hardware des Geräts konfiguriert, der Speicher initialisiert und die Programmierumgebung vorbereitet. Ein dediziertes Programmiergerät führt anschließend die Flash-Programmierung durch. Ein Blank-Check prüft, ob der Speicher leer ist. Ist dies nicht der Fall, wird er gelöscht und anschließend in manchen Fällen eine Reprogrammierung gestartet.

Anschließend werden die Daten programmiert. Hierfür sendet die Software die Daten und schreibt sie in den Speicher. Zur Überprüfung der Programmierung werden die Daten aus dem Speicher ausgelesen und mit den originalen verglichen. Dies wird als Prüfsumme bezeichnet. So wird sichergestellt, dass keine Fehler enthalten sind.

→ MEHR LESEN



Flash-Programmierung im technischen Detail

Die Informationen (Bits) werden in einer Speicherzelle in Form von elektrischer Ladung auf einem **Floating-Gate** gespeichert. Dieser Transistor ist durch eine Oxidschicht **elektrisch isoliert**, wodurch die Informationen nicht abfließen können.

Damit Informationen gezielt gespeichert werden können, ist das Aufbringen und Entfernen von elektrischen Ladungen am Gate notwendig. Hierfür gibt es verschiedene Methoden:

Hot-Carrier-Injection

Bei der Injektion heißer Ladungsträger (**Hot-Carrier-Injection**) werden Teilchen mit hohen Spannungen auf das Floating-Gate geschossen. Sind genügend Ladungen vorhanden, wird ein „Lawinendurchbruch“ ausgelöst und sie gelangen durch die Oxidschicht. Diese Methode wird bei NOR-Speichern verwendet.

Fowler-Nordheim-Tunneling

Zum Löschen der Informationen bei NOR-Speichern oder zum Programmieren und Löschen von NAND-Speichern wird das Fowler-Nordheim-Tunneling genutzt. Hier macht man sich den Tunneleffekt zu Nutzen, der es Ladungen erlaubt, Nichtleiter zu passieren. Differenziert wird hier zwischen dem **Uniform-Tunneling** und der **Drain-Slide**. Bei jedem der Fowler-Nordheim-Tunneling-Verfahren sind nur geringe Spannungen nötig.

In-System-Programmierung (ISP)

Baustein-Programmierung versus In-System-Programmierung

In der Vergangenheit war die Baustein-Programmierung die vorherrschende Methode zum Flashen von Bauteilen wie MCUs oder Speichern. Bei diesem Ansatz wurden die Komponenten vor ihrer Integration auf der Leiterplatte (PCB) einer Programmierung unterzogen.

Doch die heutigen Fortschritte haben den Flash-Prozess neu gestaltet. Beim aktuellen Paradigma der In-System-Programmierung wird das Bauteil nach der Integration auf die PCB geflasht. Dieser Wandel ist auf die Einführung der Ladungspumpe zurückzuführen, die die Umwandlung der erforderlichen Programmierspannung im Bauteil selbst ermöglicht. Folglich ist das Anlegen einer externen Spannung nicht mehr erforderlich, wodurch Risiken für die Bauteile verringert werden.

Vorteile der In-System-Programmierung

- ⊕ Möglichkeit zum Reprogrammieren
- ⊕ Kurzfristige Softwareänderungen sowie Reperaturen
- ⊕ Reduzieren der Programmierzeiten
- ⊕ Technisch-qualitative Verbesserungen
- ⊕ Schreiben von Seriennummern und dynamischer Daten
- ⊕ Geringere laufende Kosten
- ⊕ Keine Qualitätseinbuße durch Röntgen-Prüfung oder Reflow-Löten

Ablauf der In-System-Programmierung

Früher mussten Bauteile wie Mikrocontroller für die Flash-Programmierung aus der Zielanwendung zunächst entfernt und im Programmiergerät eingesetzt werden, damit die Soft- oder Firmware in den Speicher transferiert werden konnte. Anschließend wurde der MCU erneut aus dem Programmiersystem entfernt und in die Applikation eingebaut.

→ MEHR LESEN

Heutzutage kann dieser vergleichsweise aufwändige Ablauf mittels verschiedener Methoden vereinfacht werden. Eine davon ist die In-System-Programmierung (ISP). Sie ist nicht nur kostengünstig, sondern hat wesentliche Vorteile im Vergleich zu anderen Verfahren.

Denn bei der ISP werden die Bauteile erst nach der Bestückung und dem anschließenden Testprozess programmiert. Dadurch besteht keine Gefahr für die Komponenten in Hinblick auf Qualitätseinbuße durch Prüfverfahren wie Röntgen und Löten.

Auf diese Weise bleibt die Datenerhaltung (Data-Retention) unbeeinflusst.

Bei der ISP wird zunächst das Gerät vorbereitet, anschließend die Firmware heruntergeladen und die Flash-Programmierung der Applikations-Software ausgeführt. Im Anschluss erfolgen noch Funktionstests und beispielsweise Power-Cycles.

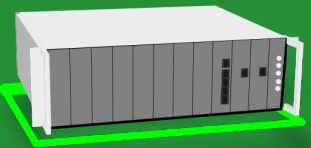
→ MEHR LESEN

1. Vorbereitung des Geräts



2. Verbindung zum Ziel

3. Firmware Download



4. Flash-Programmierung

5. Funktions-test

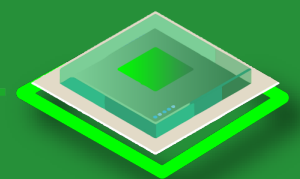


Abbildung: ISP-Ablauf

Das Herzstück der Programmierung: Der In-System-Programmer

In-System-Programmer werden zwischen PC und Device geschaltet, um die Soft- oder Firmware einer Applikationen direkt in der Linie zu programmieren. Sie ermöglichen somit volle Kontrolle über den gesamten Flash-Prozess, Flexibilität in Hinblick auf Reprogrammierung und maximale Leistung bezüglich Effizienz sowie Programmiergeschwindigkeiten.

ProMiks Multi Standard Programmer (MSP) ist die flexibelste Komponente für jegliche Flash- und Testanforderungen.

Sie können sowohl für die Flash-Programmierung über Testpunkte sowie über Automotive-Schnittstellen genutzt werden. Zudem unterstützt die MSP-Familie Boundary-Scan und ProMiks innovative Testmethode SMART ICT.



→ MEHR LESEN

Neben der MSP-Familie bietet ProMik die XDM-Serie von Programmiergeräten an.

Mit Übertragungsraten von bis zu 300 MBit/s schaffen die Programmer größtmögliche Flexibilität für die Produktionslinie. Gleichzeitig werden die Kosten auf ein Minimum gehalten. Anders als die MSP-Familie unterstützt der XDM nur gezielte Schnittstellen. So ist er mit USB- oder Automotive-Ethernet-Anschlüssen erhältlich.



→ MEHR LESEN

Die Kontrolle über den Flash-Prozess: Die Produktionssoftware

Neben performanten Programmiergeräten wird ebenfalls eine Produktionssoftware benötigt, welche den gesamten Flash-Prozess steuert und überwacht.

ProMiks FlashTask Pro ist hier die ideale Wahl. Sie ermöglicht eine intuitive Steuerung des gesamten Prozesses durch eine benutzerfreundliche Oberfläche.

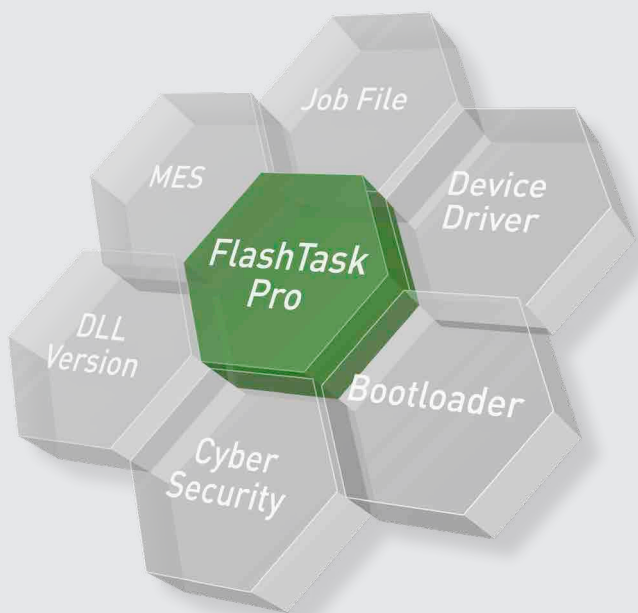


Abbildung: FlashTask Pro

Vorteile auf einen Blick

- MES-Schnittstelle
- Prozesskontrolle via JobFiles
- Rückverfolgbarkeit- und Diagnosetools
- Benutzerfreundliche Oberfläche

→ MEHR LESEN

Somit können auch komplexe Applikationen einfach realisiert werden. Zusätzlich ermöglicht die FlashTask Pro die Anbindung an das Manufacturing-Execution-System (MES).

↓ DATENBLATT

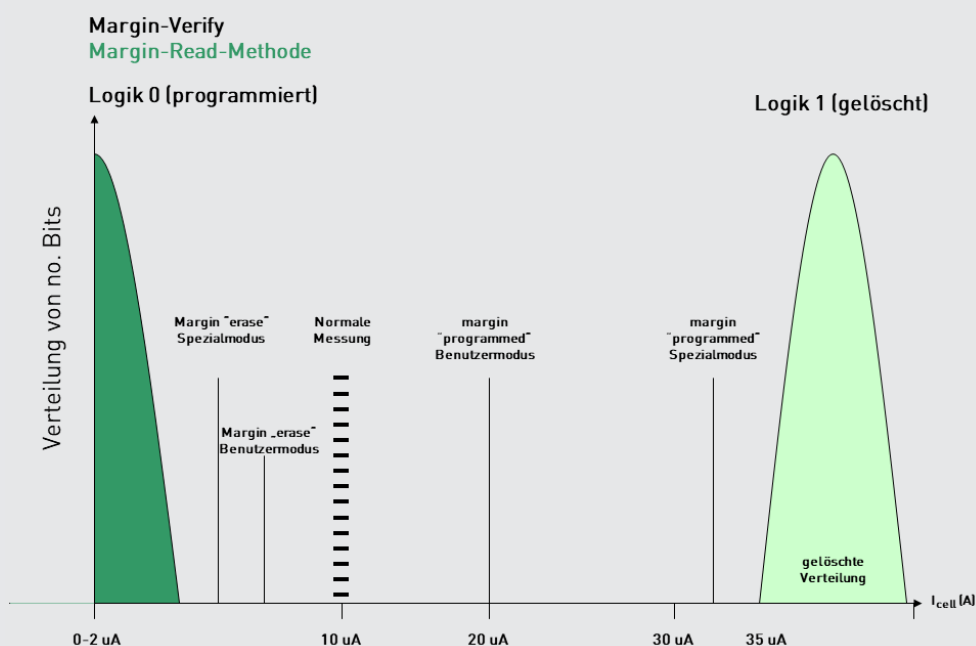
Margin-Verify entwickelt von ProMik

Indem möglichst früh geringfügig programmierte Geräte in der Produktionskette erkannt werden, kann im weiteren Verlauf nicht nur Zeit, sondern auch immense Kosten gespart werden. Der von ProMik entwickelte Margin-Verify testet die Qualität der Programmierung vollständig, um eine maximale Datenerhaltung zu garantieren. So können Produktausfälle durch das Versagen von Komponenten und unsachgemäße Programmierung minimiert werden.

ProMiks qualitativ hochwertige Programmierdienste in Verbindung des Margin-Verifys zeigen fundamentale Erfolge. So werden jedes Jahr Millionen von Geräten mit einer Fehlerrate von 0 PPM programmiert.

Use-Case

Innerhalb des Use-Cases zeigt ProMik die genaue Funktionalität des Margin-Verifys auf. Neugierig geworden? Das Anwendungsbeispiel gibt es zum Download.



[↓ DATENBLATT](#)

Abbildung: Use-Case des Margin-Verifys

ProMik: Experte für die Flash-Programmierung

Seit mehr als 25 Jahren ist ProMik als Systemlieferant in der Elektronikbranche tätig.

Das mittelständische Unternehmen bietet eine breite Palette vertrauenswürdiger Hardware- und Softwarelösungen zum Flashen und Testen einzelner Leiterplatten, Mehrfachnutzen oder gehauster Baugruppen.

→ MEHR LESEN

Besonders im Automobilsektor sowie der Industrie finden ProMiks Lösungen Anwendung. Durch Vorteile in Hinblick auf Qualität, Kosten und Effizienz überzeugen die Produkte seit Jahrzehnten.

Dank der Spezialisierung des Unternehmens auf die Flash-Programmierung sind Kunden gezielte, hochprofessionelle Lösungen garantiert, welche durch umfangreichen Support begleitet werden.

In über 7.000 Projekten konnte ProMik sein Knowhow beweisen. Komplexe Projekte liegen im Kompetenzbereich von ProMik und werden nicht als Herausforderung, sondern als Kernaufgabe betrachtet.

→ CONTACT



Use-Case: High-Speed Programmierung eines Clusters

Bei der Flash-Programmierung eines Clusters wurde der MSP2100Net und der XDM-USB von ProMik verwendet. Der ProMik-Bootloader wurde zunächst über JTAG in den i.MX8-RAM geladen, der dann die Anwendungssoftware über USB3 herunterlud. Die Anwendungsdaten wurden vom Bootloader in die eMMC und das Hyperflash übertragen, gefolgt von der Ausführung des eFusing des i.MX8.

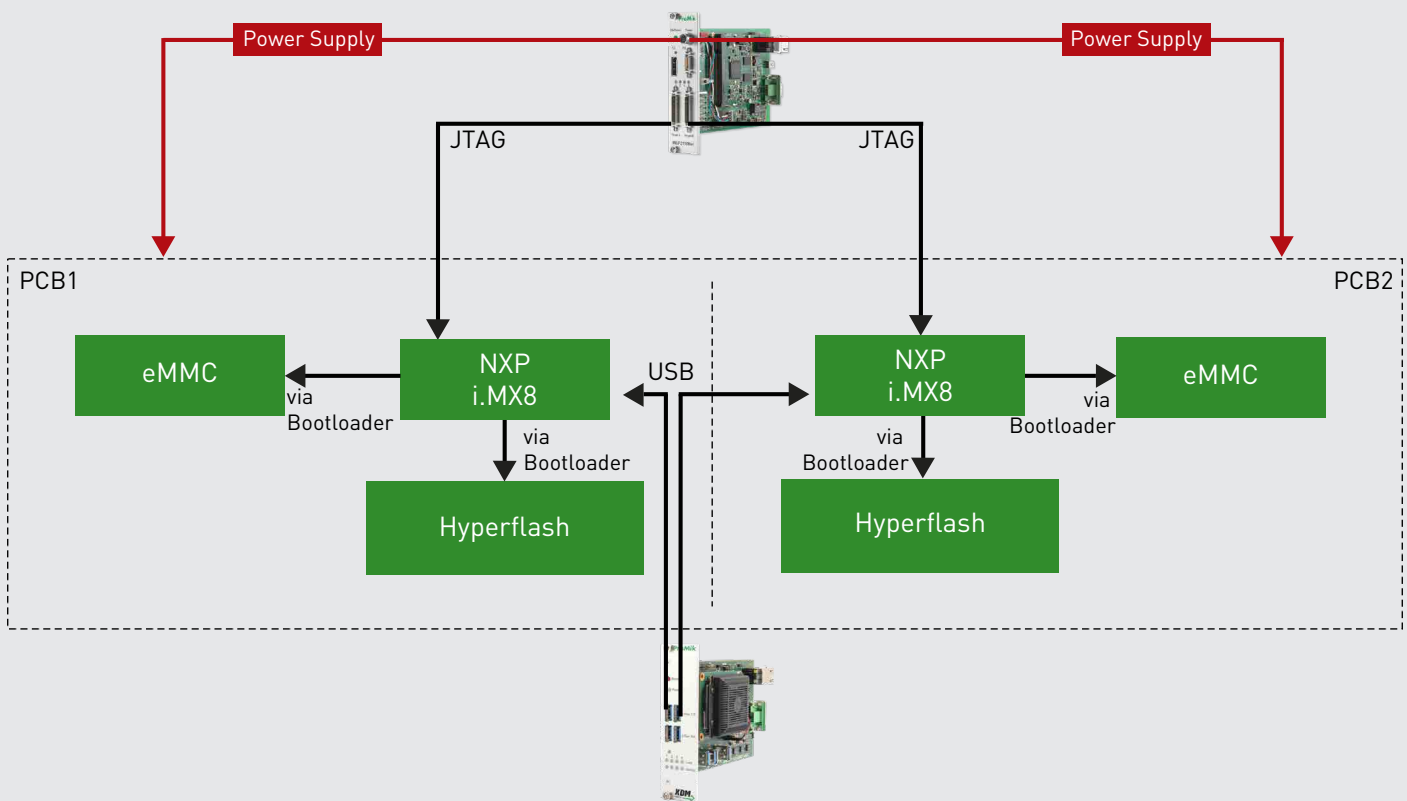


Abbildung: Programmierstrategie

Der Unterschied in der Produktion

- Maximale Leistung und höchste Qualität durch ProMiks Programmiergeräte
- Allumfassende Toolchain – anpassbar gemäß Kundenanforderungen

Erfahren Sie mehr über Flash- Programmierung

