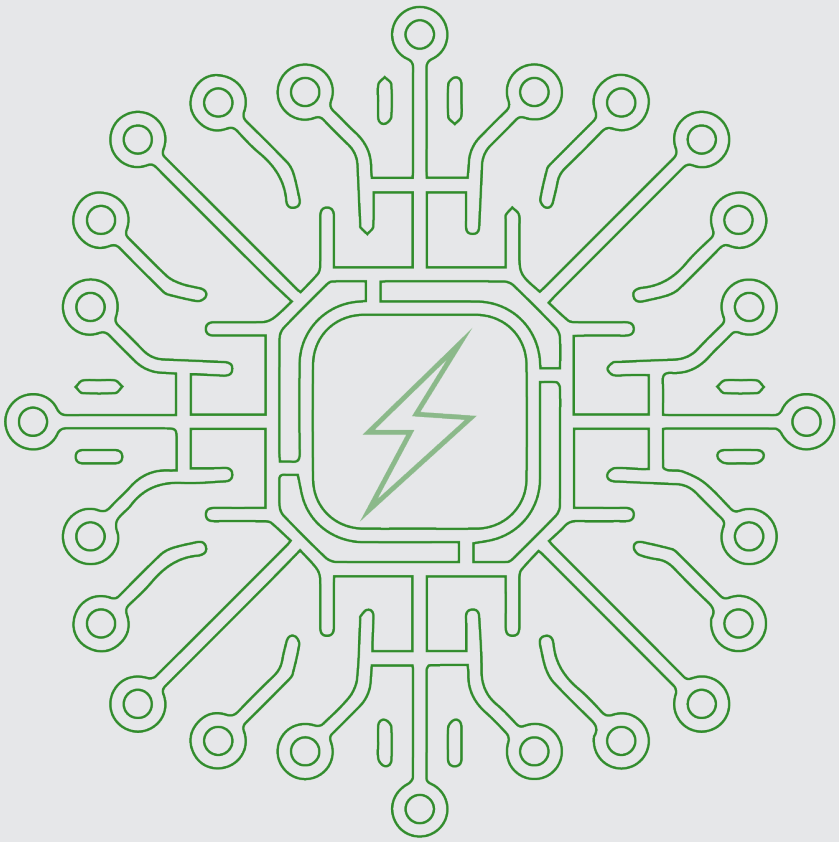
ProMik®
Trusted Solutions

hello, whitepaper

Flash Programming: Efficiency, High-Speed and Performance for Electronics



Flash programming is the most common method for the programming of individual PCBs, multiple use or housed assemblies. Read all the details here!

Content

3
Relevance

4
Structure of a Microcontroller (MCU)

5
Flash Memory: Functionality

6
General Description of Flash Programming

7
Flash Programming in Technical Detail

8
In-System Programming (ISP)

10
The Heart of Programming: The In-System Programmer

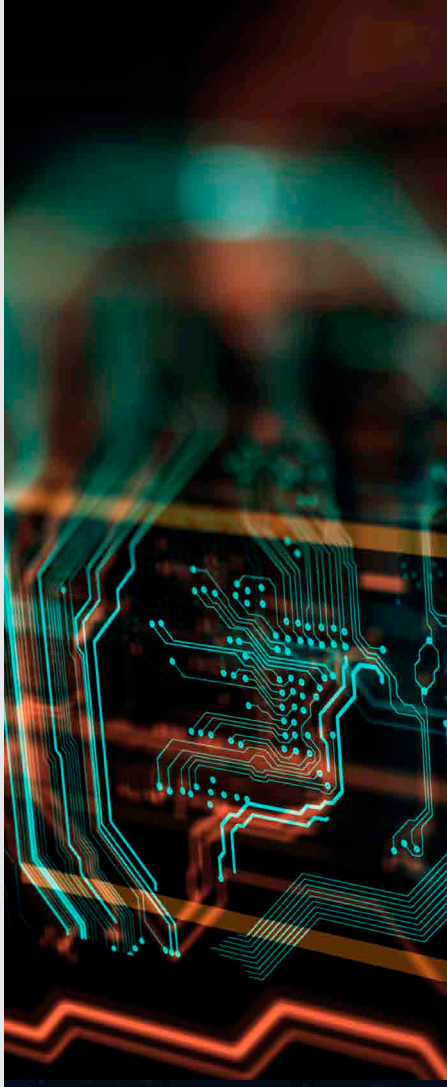
11
Control over the Flash Process: The Production Software

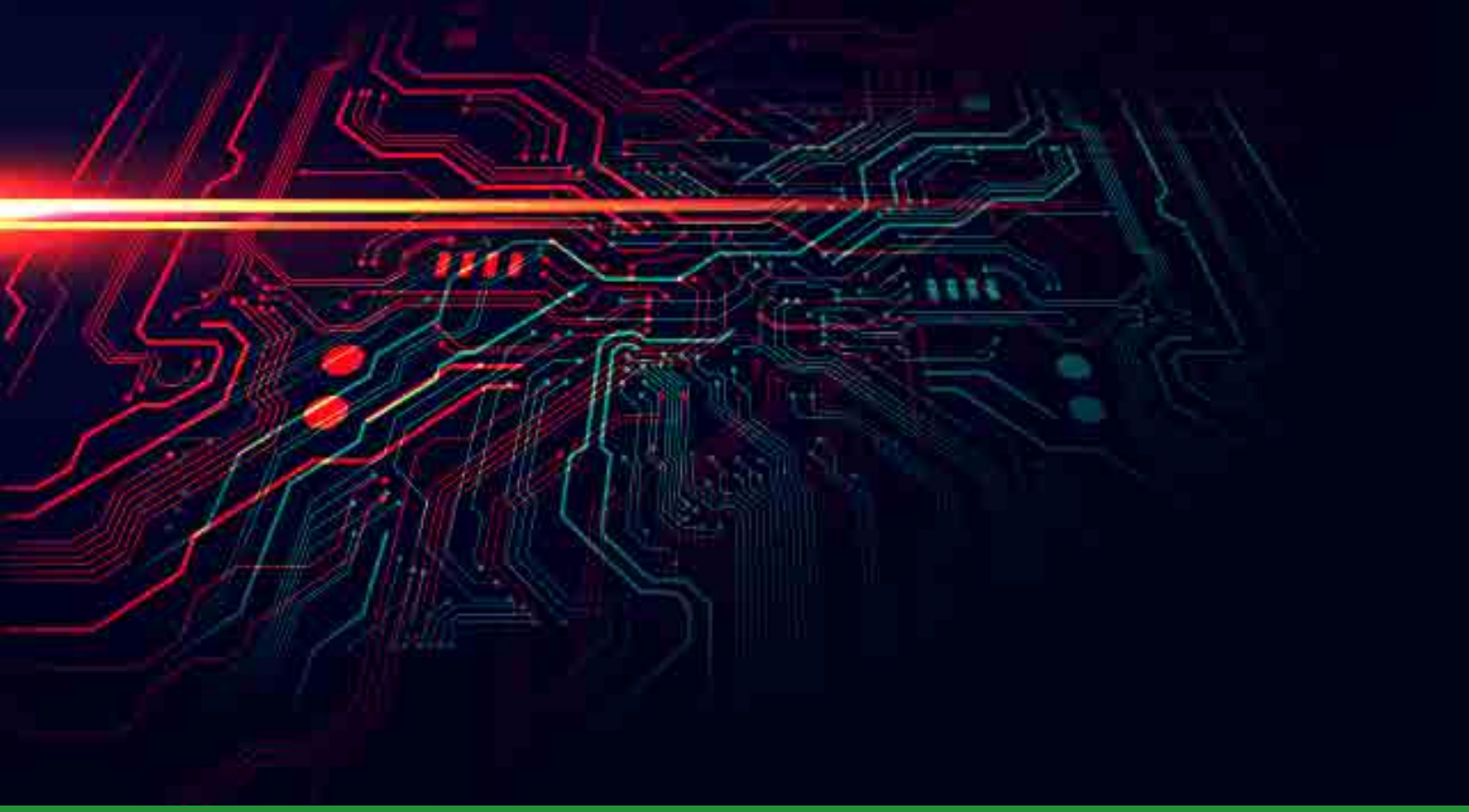
12
Margin Verify developed by ProMik

13
ProMik: Expert for Flash Programming

14
Use Case: High-Speed Programming of a Cluster

15
Get to know more about Flash Programming





Relevance

Flash programming has attained heightened significance in tandem with the escalating complexity of technologies. Evolving from its previous role within the in-circuit test (ICT), flash programming has now become a standalone production step. Its merits, distinguishing it from other non-volatile programming methods, encompass the capacity for reprogramming, fast read access, and exceptional data density.

Flash programming is used in almost all industries. In-system programming, which describes the programming after the board assembly, is particularly suitable in areas where high volumes and complex applications are required. An ideal example of this is the automotive industry.

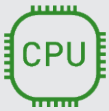
Since flash programming is a highly technical process, it requires the right know-how. In this realm, aligning with professionals like ProMik ensures customers are impeccably equipped.

Structure of a Microcontroller (MCU)

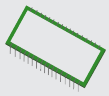
A microcontroller (MCU) is a semiconductor chip, which contains both a processor and peripherals. Its task is the measurement, control and regulation of applications.
The functionality is as follows: The component detects external signals via communication protocols and reacts to them.

MCUs are divided into different **core types** (8-bit, 16-bit and 32-bit) as well as **architectures** (e.g. Reduced Instruction Set Computer (RISC) or Acorn RISC Machine (ARM)).

Microcontrollers also have different components. These include the central processing unit (CPU), various memories, I/O connections and peripherals.



CPU: central processor



Memory: at least one RAM and one ROM or **flash memory**

I/O connections

Peripherals: e.g. **communication** or **debugging interfaces**

Flash Memory: Functionality

Flash memories are non-volatile memories that differ from ROM memories since they are **reprogrammable**. By using high voltages, data can be erased and rewritten. Flash memories are therefore much **more flexible**, have **high data transmission rates** and **faster reaction times** than ROM memories.

A flash memory consists of different elements – among them a control - and floating gate. The floating gate stores the information and is seperated from the rest of the MCU by an oxide layer. Additionally, the memory contains a source where charges arrive and a drain where they flow off again. Details about the function of the components can be read in the technical detail.

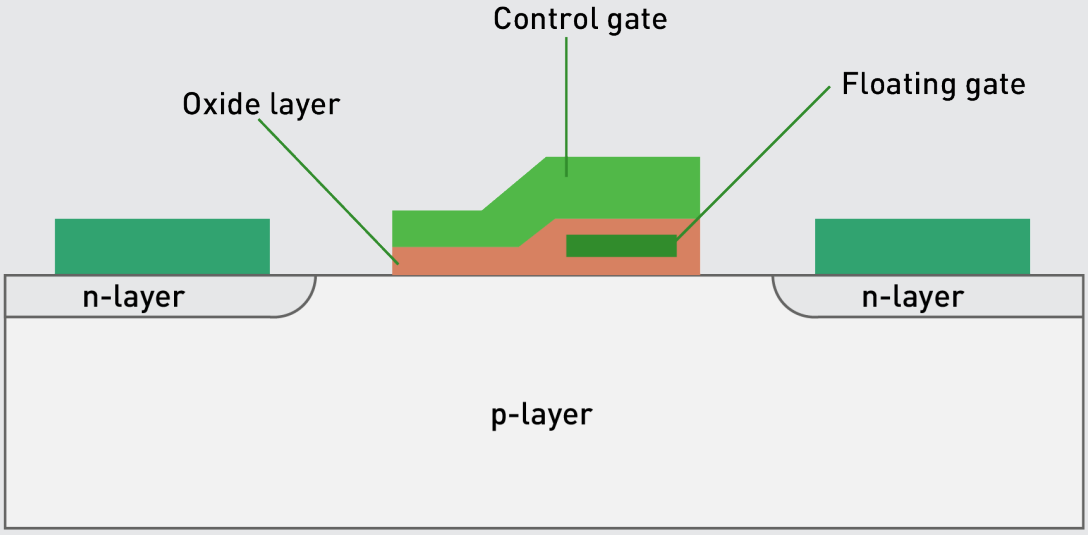


Figure: Flash memory structure

Flash memories are divided into Not AND (**NAND**) and Not OR (**NOR**) **memories**.

In case of NAND memories the memory cells are serially connected. They are used as mass storages and reduce the required space.

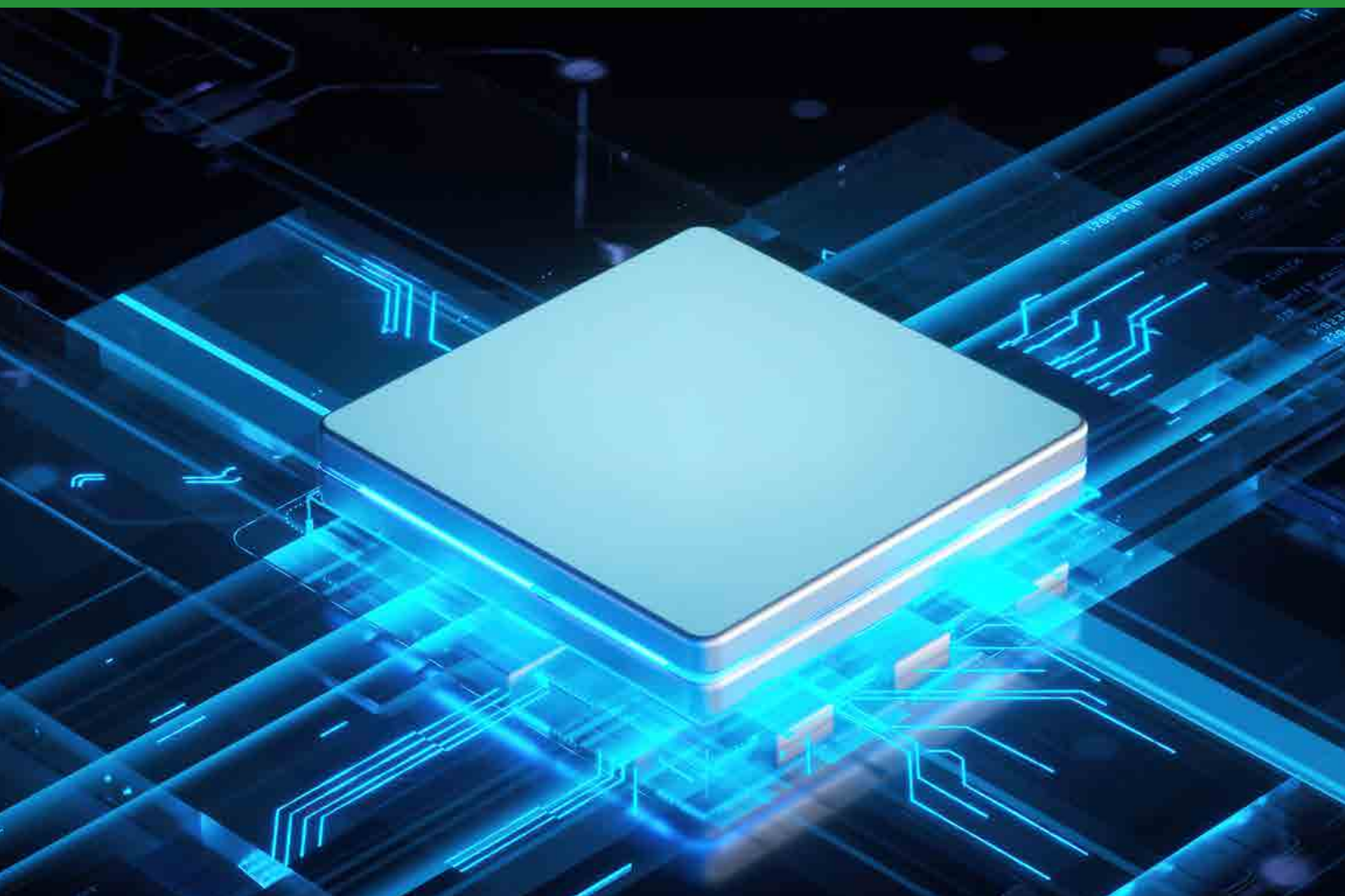
The memory cells of NOR memories are switched parallely. This enables direct access to the data, however, more space is needed.

General Descrpition of Flash Programming

First, the hardware of the device is configured, the memory is initialized and the programming environment is prepared. A dedicated programming device then carries out the flash programming. A blank check verifies whether the memory is empty. If this is not the case, it is deleted and in some cases reprogramming is then executed.

During the next step, the data is programmed. For this, the software sends data and writes it in the memory. To check the programming, the data is being read out of the memory and compared to the original. This is called checksum. Like this, it can be ensured that there are no errors included.

READ MORE



Flash Programming in Technical Detail

The information (bytes) is stored in a memory cell in the form of electrical charge on a **floating gate**. This transistor is **isolated electrically** with an oxide layer, which prevents the information from flowing away.

In order to store information selectively, the application and removal of electrical charges on the gate is necessary. To do so, there are multiple methods.

Hot Carrier Injection

During the injection of hot carriers, particles are being shot with high voltages at the floating gate. As soon as a sufficient amount of charges is available, an avalanche break is triggered and they get through the oxide layer. This method is used with NOR memories.

Fowler Nordheim Tunneling

To delete the information from NOR memories or to program and erase NAND memories, the Fowler Nordheim Tunneling is applied. Here the tunnel effect is utilised which allows charges to cross non-conductors. A distinction is made between Uniform Tunneling and the Drain Slide. With all of the Fowler Nordheim Tunneling methods only low voltages are required.

In-System Programming

Device Programming versus In-System Programming

Historically, device programming stood as the prevailing method for flashing components like MCUs or memories. In this approach, components underwent programming before their integration onto the printed circuit board (PCB).

Yet, contemporary advancements have reshaped the flash process. In the current paradigm of in-system programming, the component undergoes flashing post-integration onto the PCB. This transformative shift is attributed to the introduction of the charge pump innovation, enabling the conversion of the necessary programming voltage within the component itself. Consequently, external application of voltage is no longer requisite, mitigating risks to the components during the process.

Advantages of In-System Programming

- Possibility of reprogramming
- Short-term software changes as well as repairs
- Reduction of programming times
- Technically qualitative improvements
- Writing of serial numbers and dynamic data
- Lower running costs
- No quality losses through x-ray tests or reflow soldering

Process of In-System Programming

In the past, components such as microcontrollers had to be first removed from the target application and inserted in the programming device for flash programming so that the software or firmware could be transferred to the memory. Then the MCU was again removed from the programming system and inserted into the application

READ MORE

Nowadays, this comparatively complex process can be simplified using various methods.
One of these is in-system programming (ISP). It is not only cost-effective, but also has significant advantages compared to other methods.
With ISP, the components are only programmed after assembly and the subsequent test process. This means that there is no risk to the components in terms of quality losses due to test procedures like x-ray and soldering.
Like this, data retention remains unaffected.

During the ISP, the device is first prepared, then the firmware is downloaded and the flash programming of the application software is carried out. This is followed by functional tests and for example power cycles.

READ MORE

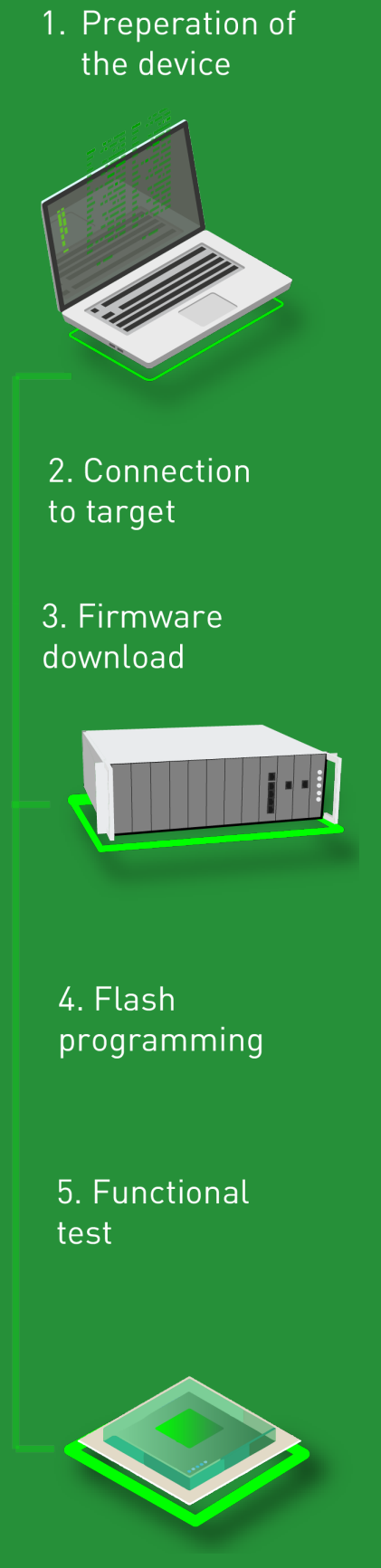


Figure: ISP process

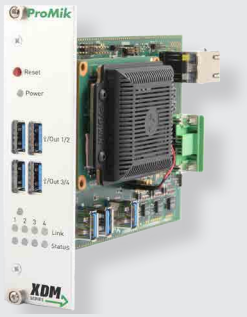
The Heart of Programming: The In-System Programmer

In-system programmers are connected between the PC and the device to program the software or firmware of an application directly in-line. They allow full control over the entire flash process, flexibility in terms of reprogramming and maximum performance regarding efficiency and programming speeds.

ProMik's Multi Standard Programmer (MSP) is the most flexible component for any flash and test requirements.
They can be used for flash programming via test pads as well as automotive interfaces. In addition, the MSP family supports boundary scan and ProMik's innovative SMART ICT test method.

READ MORE

In addition to the MSP family, ProMik offers the XDM series of programmers.
With transmission rates of up to 300 MByte/s, the programmers create the greatest possible flexibility for the production line. At the same time, costs are kept to a minimum. Unlike the MSP family, the XDM supports only targeted interfaces. Thus, it is available with USB or automotive Ethernet connections.



READ MORE

Control over the Flash Process: The Production Software

In addition to high-performance programming devices, a production software is also needed to control and monitor the entire flash process. ProMik's FlashTask Pro is the ideal choice here. It enables intuitive control of the entire process through a user-friendly interface.

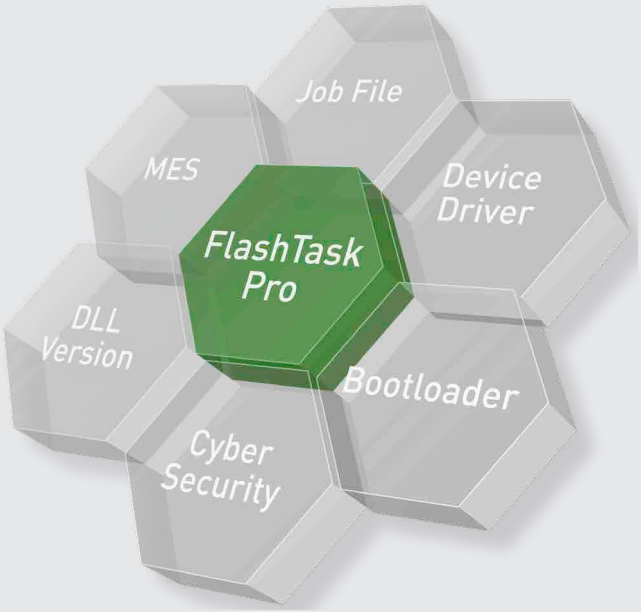


Figure: FlashTask Pro

Advantages at a glance

- MES interface
- Process control via JobFiles
- Traceability and diagnostic tools
- Operator-friendly interface

READ MORE

As a result, even complex applications can be easily realised. In addition, FlashTask Pro enables the connection to the Manufacturing Execution System (MES).

DATASHEET

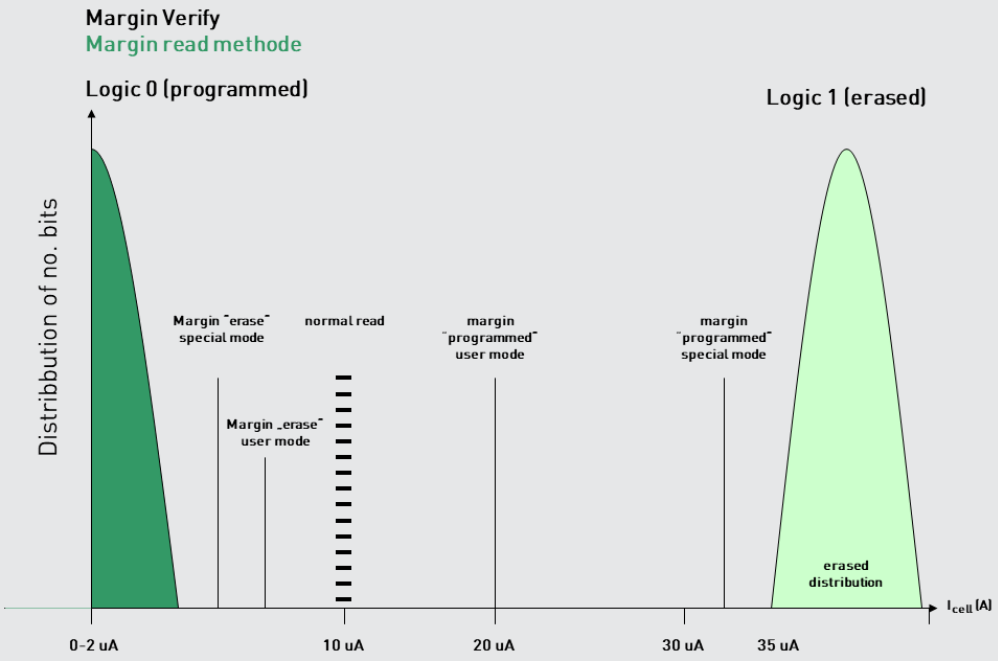
Margin Verify developed by ProMik

By detecting marginally programmed devices in the production chain as early as possible, not only time but also immense costs can be saved further down the line. The Margin Verify, developed by ProMik, fully tests the quality of the programming to guarantee maximum data retention. This minimizes product failures due to component failure and improper programming.

ProMik's high quality programming services in conjunction with Margin Verify have shown fundamental success. Millions of devices are programmed every year with an error rate of 0 PPM.

Use Case

Within this use case, ProMik shows the exact functionality of the Margin Verify. Curious? Download the application example.



DATASHEET

Figure: Use Case Margin Verify

ProMik: Expert for Flash Programming

For more than 25 years, ProMik has been active as a system supplier in the electronics industry.

The medium-sized company offers a wide range of trusted hardware and software solutions for flash programming and testing individual PCBs, complete boards or housed assemblies.

READ MORE

ProMik's solutions are used particularly in the automotive sector and in the industrial sector. They have convinced since decades with their advantages in terms of quality, costs and efficiency.

Thanks to the company's specialisation in flash programming, customers are guaranteed targeted, highly professional solutions that are accompanied by holistic support.

ProMik has been able to prove its know-how in over 7,000 projects. Complex projects are within ProMik's realm of expertise, not regarded as a challenge but as their core duty.

CONTACT

Use case: High-Speed Programming of a Cluster

For the flash programming of a cluster, ProMik's MSP2100Net and XDM-USB were used. The ProMik Bootloader was first downloaded into the i.MX8 RAM via JTAG, which then downloaded the application software through USB3. The application data was then transferred into the eMMC and Hyperflash by the bootloader, followed by the execution of the eFusing of the i.MX8.

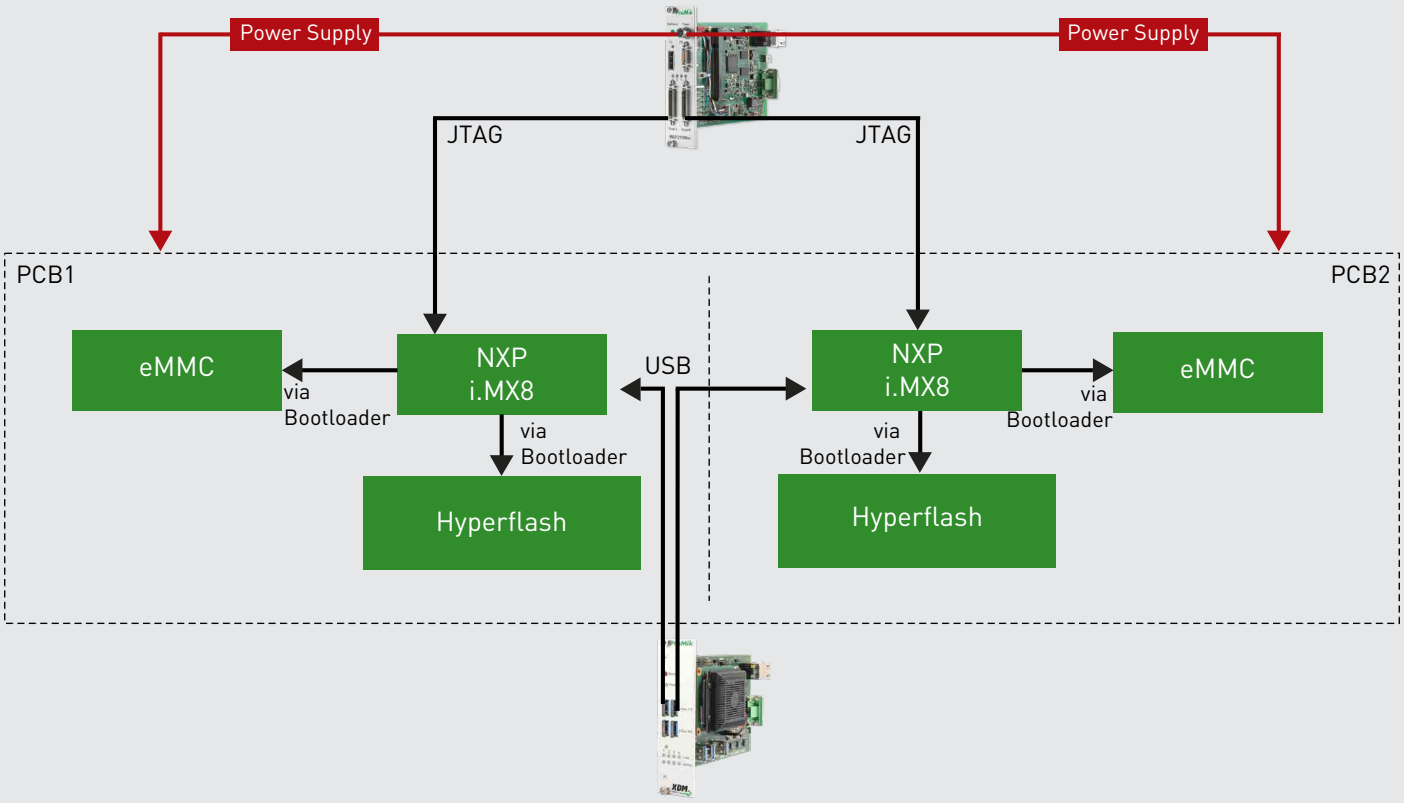


Figure: Programming strategy

The difference in production

- Maximum performance and highest quality with ProMik's programmers
- Holistic toolchain – adaptable to customer requirements

Get to know more about Flash Programming

ProMik®
Trusted Solutions