

## Flash-Programmierung in der Produktion

# Zeit ist Geld – Programmierzeit auch

31. Januar 2022, 6:00 Uhr | [Harry Schubert](#)



**Die Software für Embedded-Systeme muss in der Produktion sicher, zuverlässig und schnell in den Flash-Speicher übertragen werden. Im Interview erklärt Produktmanager Peter Orda von Segger Microcontroller, die Vorteile eines Programmiersystems mit zentralem Hub und verteilten Programmiergeräten.**

In der Produktion von Embedded-Geräten ist die Programmierung der Mikrocontroller-Firmware nur einer von vielen Arbeitsschritten. Die Flash-Programmierung kann aber durchaus ein zeitaufwendiger Schritt sein. Um eine hohe Effizienz der Produktion zu erzielen, muss die Programmierzeit somit minimiert werden. Segger Microcontroller hat ein Programmiersystem mit einem zentralen Hub und verteilten kompakten Programmiergeräten entwickelt, das eine hohe Programmiergeschwindigkeit ermöglicht.

## ? Herr Orda, was hat Sie zur Entwicklung eines Programmiersystems für den Einsatz in Produktionslinien motiviert?

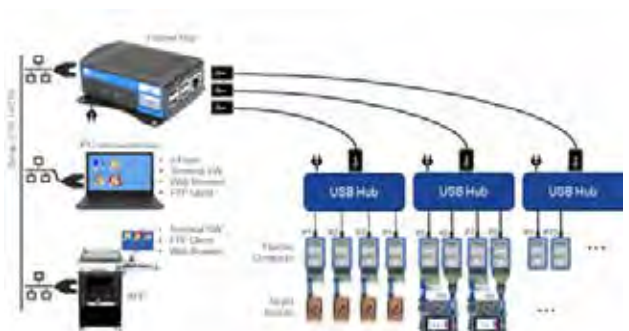
! **Peter Orda:** Wir hatten bei Segger bereits eine Lösung für die Gang-Programmierung mit dem »Flasher ATE« im Angebot. Aus Gesprächen und Anfragen von Kunden haben wir gelernt, dass die Konzeption des ATE nicht immer vollständig den Bedürfnissen der Kunden entspricht. Als Hauptproblem wurde uns die räumliche Begrenzung durch die Leitungslängen des »Flasher ATE« genannt. Viele Kunden benötigen eine Lösung, bei der ein Programmieradapter möglichst nahe am Target platziert werden kann, um maximale Programmiergeschwindigkeiten zu erreichen, andererseits die Abstände zwischen den ATE möglichst flexibel sein kann.



Peter Orda ist als Produktmanager für die gesamte Flasher-Familie bei Segger Microcontroller verantwortlich: »Unsere Programmierzeit ist möglichst nahe am technisch möglichen Minimum.«

Wir sind stolz, nun mit dem Flasher Compact (FC) genau solch eine Lösung vorstellen zu können. Er ist klein genug, um nahezu direkt am Target untergebracht zu werden, und ermöglicht damit sehr kurze Leitungslängen der Programmierschnittstelle. Auf der anderen Seite haben wir mit der USB-Schnittstelle eine schnelle Standardschnittstelle mit genügend großer Reichweite zur Verfügung, um nahezu alle denkbaren geometrischen Verteilungen von Testplätzen zu unterstützen.

Dabei ist der FC ein typischer Segger-Flasher mit der bekannten hohen Geschwindigkeit, die in den meisten Fällen sehr nahe an das theoretische Maximum heranreicht. Die Konfigurationsprogramme »J-Flash« und »Universal Flash Loader Configurator« sind bereits seit Jahren sowohl für Windows, MacOS und die meisten Linux-Distributionen für Intel verfügbar.



Übersicht Flasher Hub und Flasher Compact  
Bis zu 24 Flasher Compact lassen sich am Flasher Hub parallel betreiben, um den Flash-Speicher von Embedded-Systemen zu programmieren.  
© Segger Microcontroller

Der Flasher Hub ermöglicht die zentrale Steuerung von bis zu 24 FC, zusätzlich zu den durch den FC unterstützten Einzelplatz-Modi, wie von unseren anderen Flasher-Produkten bekannt. Damit können bis zu 24 Targets parallel mit individuellen Firmware-Images programmiert werden, Stand-alone – nur mittels Tastendruck bedient – oder automatisiert gesteuert über ein CLI per UART, inkl. dedizierten Leitungen für Ready/Error oder TELNET.

Im Flasher Hub ist zusätzlich ein Webserver integriert, als weitere Kommunikationsmöglichkeit, neben dem CLI per UART und TELNET. Damit lässt sich der Verbund von bis zu 24 FC komfortabel steuern und der Status abfragen.

Über den FTP-Server im Hub können die Firmware-Images und die zugehörigen Konfigurationsdateien, darunter auch Seriennummern und Patch-Konfiguration, eines oder auch mehrerer Projekte auf die Flasher transferiert werden. So können aber auch die Protokoll-dateien des Flashers abgeholt werden.

### **? Was ist konkret der Vorteil des Flasher Compact gegenüber anderen Programmiergeräten?**

**! Orda:** Ich habe durch Gespräche mit Kunden, Kollegen und durch eigene Erfahrung gelernt, dass gerade Programmier- und Testzeiten das Nadelöhr in einer Produktion sein können. Bei einem geplanten Ausstoß kann u.U. eine Sekunde mehr oder weniger darüber entscheiden, ob ein zusätzlicher Testplatz nötig ist. Und ein zusätzlicher Testplatz ist eine nicht unerhebliche Investition. Deshalb wollen wir unsere Kunden unterstützen, indem wir immer nach Möglichkeiten suchen, damit die Programmierzeit möglichst nahe am technisch möglichen Minimum ist.

Neben einer schnellen Programmierschnittstelle mit sauberen Signalen und kurzen Leitungslängen als physikalische Voraussetzung verwenden wir, wenn immer möglich, die schnellsten Programmiermodi der Targets. Bei vielen Targets wird ein speziell dafür erstellter Code in den RAM geladen, der die Programmierung übernimmt. Damit entfällt ein Teil des Protokoll-Overheads, der für eine direkte Programmierung über das ISP nötig wäre.

Um auch die Zeit, die zur Programmierung des Flashs benötigt wird, sinnvoll zu nutzen, lädt unser RAM-Code im Turbo-Mode währenddessen Daten in den RAM, damit möglichst ohne Totzeiten programmiert werden kann. Dabei werden, wann immer möglich, Datenmengen in der verfügbaren RAM-Größe bewegt, um auch hier den Protokollaufwand zu minimieren.

**Es gibt ja tausende unterschiedliche Mikrocontroller und Speicher-ICs. Wieviele davon unterstützt Segger mit dem Flasher Compact und was kann ein Kunde machen, wenn er andere Bausteine verwenden will?**

**! Orda:** Ja, und leider – oder zum Glück – will jeder Speicher anders angesprochen werden. Selbst innerhalb einer Familie gibt es subtile Unterschiede, so dass es meist nicht damit getan ist, die Adressbereiche anzupassen. Außerdem werden durch gestiegene Sicherheitsanforderungen die Hürden, um überhaupt auf den Speicher zugreifen zu können, immer höher. Deshalb ist es uns leider nicht möglich jeden neu erscheinenden Speicher sofort zu unterstützen. Dazu kommt die zum Teil sehr dürftige Dokumentationslage. Hier sind wir auf die Bereitschaft des Herstellers angewiesen, uns zeitnah Dokumentation und Support zu gewähren. Bei einigen Targets sind wir gezwungen reverse engineering zu betreiben, da wir keine Informationen bekommen oder diese für unsere Zwecke unbrauchbar sind.

Wir versuchen populäre Reihen nach und nach komplett zu unterstützen. Allerdings sind uns hier schon aufgrund der Vielzahl von Herstellern, Reihen und neu vorgestellten Targets die bei uns angestellten Entwickler mehr als gut ausgelastet.

Wenn ein Kunde zeitnah für ein Projekt noch nicht unterstützten Speicher verwendet, tun wir unser Möglichstes, um eine Unterstützung durch unsere Flasher-Produkte dafür zur Verfügung zu stellen. Dabei können jedoch – je nach Komplexität des Bausteins – Kosten anfallen, die wir zu – so denke ich – recht fairen Bedingungen dem Auftraggeber in Rechnung stellen müssen.