

# Kapitel 2

## Hauptschleife

Die Hauptschleife (Run-Loop) ist die einfachste Methode, ein embedded System zu betreiben. In dieser Schleife werden nacheinander Unterprogramme aufgerufen, die zur Kommunikation, Steuerung oder für andere Aufgaben dienen. Tatsächlich gibt es das Hauptschleifenkonzept auch als Prinzip in Betriebssystemen. Apple hat in seinen Dokumentationen zum iOS-Betriebssystem (iPhone, iPad) lange die Run-Loop gegen Multithread-Programmierung verteidigt. Wir nutzen das Hauptschleifenkonzept, wenn für unser System keine Betriebssysteme erhältlich sind oder sich nicht leicht bauen lassen oder wenn der Speicher zu knapp für ein OS ist. Abgesehen von möglicher Interrupt-Programmierung ist dieses Hauptschleifenprogramm nicht nebenläufig, sondern streng sequenziell. Das reicht für überschaubare Probleme oft aus und ist sehr kompakt. Am Beispiel dieser Nische können wir gut studieren, wie wir ein solches System zum ersten Mal in Betrieb nehmen können. Für einen ersten Start eines Prozessorsystems bei der Inbetriebnahme braucht man einen Minimal-Lader, der in einer Schleife ein Programm von einer Schnittstelle (z. B. seriell) in den RAM-Speicher lädt und dort schließlich startet. Läuft dieser Bootloader, dann können alle weiteren Funktionen wie zum Beispiel die Flash-Programmierung, weitere HW-Settings etc. programmiert werden. Anschließend können wir ein erstes Hauptschleifenprogramm starten und dieses schrittweise aufbauen und mit Unterprogrammen ergänzen.

### 2.1 Serielles EEPROM

In früheren Systemen wurde dieser erste Start oft mit einem EEPROM (Boot-ROM) realisiert, das zum Programmieren ausgebaut werden musste oder mit Pin-Adaptoren in der Schaltung programmiert wurde. Mittlerweile wurde dieses Vorgehen durch den Einsatz der JTAG-Adapter abgelöst, siehe unten.

## 2.2 Emulator

Alternativ gab und gibt es Emulatoren, die man statt des Prozessors in die Platine einsetzt. Im Emulatorkopf befindet sich ein verdrahteter Spezialprozessor, der den Zugang zu allen Registern und Adressbereichen erlaubt. Die Verdrahtung wird dann zu einer weiteren Schaltung im PC geführt, auf der dann die Emulator-Logik und -Ansteuerung läuft. Auch ein Single-Stepping mit einem Debugger wird in Hardware unterstützt. Von Nachteil ist oft, dass man eine spezielle Entwicklungsplatine braucht und nicht in Serien-Hardware Fehler suchen kann. Weiterhin hat dieser Aufbau durch die zusätzliche Verkabelung oft nicht das Serien-Timing und naturgemäß eine andere Einbaugeometrie.

## 2.3 JTAG-Adapter

Ein JTAG-Adapter ist ein Stecker, mit dem man einem IEEE-Standard entsprechende elektronische Hardware zum Testen und Debuggen adaptieren kann. Der Standard definiert auch eine Beschreibungssprache, in der mit dem Boundary-Scan-Verfahren integrierte Schaltungen im eingelöteten Zustand auf Funktion getestet werden können. Insbesondere für die Inbetriebnahme von Prozessorsystemen hat sich JTAG mittlerweile etabliert. Jedes übliche Prozessorboard sieht einen 10-poligen JTAG-Stecker vor. Es können Debugger adaptiert werden, die auf Register-Ebene vollständig debuggen können, und angeschlossene Flash-Bausteine sind aus Prozessorsicht vollständig programmierbar. Für die effiziente Inbetriebnahme neuer Boards sind ein JTAG-Interface und ein passender Adapter/Debugger/Flasher übliche Voraussetzung. Oft schreibt man ein erstes Bootloader-Programm mit dem JTAG-Adapter in den Flash-Baustein und lädt mit diesem anschließend über eine Schnittstelle alle weiteren Programme ins RAM, um sie dort auszuführen. Auch professionelle Bootloader, die viele Monitor- und Debug-Funktionen mitbringen, lädt man über JTAG das erste Mal in den Flash-Speicher.



<http://www.springer.com/978-3-642-23995-3>

Embedded Technologies

Vom Treiber bis zur Grafik-Anbindung

Wietzke, J.

2012, XXVI, 321 S. 114 Abb., 20 Abb. in Farbe.,

Hardcover

ISBN: 978-3-642-23995-3