

# Das sind die Embedded Trends 2018

12 Februar 2018 | **Insight Zühlke, Internet of Things** | [Edgar Holleis](#)

**Lesezeit:** 5 Minutes

Aktuelle Trends folgen nicht jährlichen Mustern. Auch nicht im Embedded Bereich. So sind Technologien, die bereits vergangenes Jahr den Diskurs der Fachwelt bestimmten, auch heuer relevant. Wir zeigen Ihnen Embedded Trends, mit welchen sich unsere Experten/innen nicht nur thematisch beschäftigen, sondern auch in Projekten praktische Erfahrungen sammeln.

## 1. Internet of Things (IoT)

Trotz seiner Höhen und Tiefen ist das Internet der Dinge (Internet of Things, IoT) auf der Metaebene immer noch das Zugpferd der Embedded-Arena. In den nächsten fünf Jahren werden fast 6 Billionen US-Dollar für IoT-Lösungen ausgegeben (BI Intelligence Bericht), was bedeutet, dass mit einer massiven Zunahme an digital verbundenen Geräten zu rechnen ist und die Nachfrage nach Low-Power, vernetzten und Sensor-eingebetteten Geräten steigen lässt. Weitere Artikel zu IoT auch in unserem [Zühlke Blog](#).

## 2. Neue Prozessortechnologien

Auch in diesem Jahr werden neue Fortschritte in der Prozessortechnologie die Embedded- und IoT-Anwendungen vorantreiben. Insbesondere ARM und seine Partner haben erhebliche Investitionen in IoT-Lösungen getätigt. 2018 könnte das Jahr sein, wo [Cortex M23](#) und [M33](#) basierte Designs auf verfügbar werden. Das sind gute Neuigkeiten für die Sicherheit von Embedded Devices!

Entwickler sollten jedoch ARM-Alternativen nicht aus den Augen verlieren. MIPS- und x86 verteidigen zur Recht ihre jeweiligen Nischen. Für ASICs und FPGAs ist außerdem ein neuer Spieler hinzugekommen: RISC-V.

## 3. IoT-Betriebssysteme und -Plattformen

Welche IoT-Plattform löst mein Problem am schnellsten, elegantesten und kostengünstigsten? Ist Mbed OS die derzeit beste Open Source RTOS Plattform oder nicht? Diese Frage bekommen unsere Experten/innen immer wieder gestellt und haben es sich [Mbed OS](#) genauer angeschaut:

[Mbed OS](#) ist derzeit die größte RTOS-Plattform am Markt mit einer immer noch größer werdenden Community. Die Entwicklung wird dabei neben der Community direkt von Arm getrieben. Dadurch wird eine professionelle Lösung gewährleistet. Dies hat aber auch zur Folge, dass nur Controller und Prozessoren auf Arm Basis unterstützt werden.

## Mbed OS im Check-up

### Die Vorteile

- **Hohe Portabilität** durch eine gute Abstraktion der HAL durch Treiber. Dabei wird die *Thread Security* bereits durch die einzelnen Treiber abgedeckt.
  - Unterstützung von **Low Power Features** wie dynamische Sleep Modi, Low Power UART und Low Power Timer, sofern der Controller diese unterstützt.
  - Breite Unterstützung von **externen Bauteilen** (u.a. Flash Devices, Sensoren, Schnittstellenbausteinen) sowie **internen Libraries** (File Systems, Buffers, etc) welche ständig wächst und gut gewartet wird.
  - Derzeit über **100 unterstützte Devices** von **13 Herstellern** u.a. Silicon Labs, STMicroelectronics, NXP und Renesas. Darunter fallen auch Devices mit Wireless Technologies wie Bluetooth LE, WLAN, ZigBee und 6LoWPAN bei welchen eine nahtlose Integration bzw. Abstraktion der Schnittstellen in das OS erfolgt.
  - Sehr **geringe Entwicklungszeit** bis zur ersten Applikation. Die Steuerung der Blinkfrequenz einer LED über den die UART des PC hat eine Entwicklungszeit von unter einer Stunde bei ohne Vorwissen von Mbed oder über den verwendeten Controller vorangegangener Programmiererfahrung.
  - **Unabhängigkeit** gegenüber der IDE und des Compilers. Mbed bietet über das Command Line Interface (CLI) und den Online Compiler eine Exportmöglichkeit des OS auf die meisten IDEs (u.a. IAR, Keil, Eclipse, VSCode) und Compiler.
  - Fokus der Entwicklung kann auf **Applikation** liegen:  
Aufgrund des guten Abstraktionslevels von Mbed kann der Fokus viel mehr auf die Entwicklung der Applikation gelegt werden.
  - **Große Community**, welche schnell Bugs findet und diese auch zeitnahe beseitigt, sowie immer wieder **neue Features**, Libraries und Treiber für externe Komponenten hinzufügt.
  - Einfache und **schnelle Integration** des Mbed OS in die Mbed IOT Plattform.
- Die Nachteile
- **Hoher Speicherverbrauch** für ein mikrocontrollerorientiertes RTOS. Das einfache „Blinky“ Example (periodisches Blinken einer LED) braucht zwar nur ca. 5 Zeilen Applikationscode, jedoch hat das Programm einen Flashspeicherbedarf von bereits 80kB.
  - Low Power Features sind **nicht** immer **ausgereift**. Die speziellen Low Power Features der einzelnen Hersteller werden nur teilweise unterstützt, sofern sie vom jeweiligen Hersteller selber in die Mbed API integriert wurden.
  - Aus dem Quellcode ist **nicht ersichtlich** welche **Features** unterstützt werden. Dadurch, dass Mbed eine große Anzahl an Devices unterstützt ist es aus dem Code heraus nur schwierig ersichtlich, (durch define-Verschachtelungen) ob und welche Features gerade unterstützt werden (z.B. UART und Low Power UART).
  - Bei der Unterstützung von devicespezifischen Features ist man auf die Implementation durch

den **jeweiligen Hersteller** in die Mbed API **angewiesen**.

Zusammenfassend kann gesagt werden, dass die Vorteile von Mbed OS überwiegen und auch ein hohes Potential für die kommenden Jahre aufweisen, vor allem aufgrund der Portabilität und der Entwicklungsbeschleunigung durch die große Anzahl an Treibern.

#### **4. Blockchain**

Blockchain bietet Embedded-Programmierern noch relativ unbekanntere Anwendungsmöglichkeiten. Doch eines ist sicher: auch unsere Experten/innen werden sich 2018 intensiv mit dem Thema Blockchain und Embedded beschäftigen.

Ein mögliches Einsatzszenario ist beispielsweise die Verwendung von Blockchain in der Supply Chain Visibility. Durch die Verwendung eines shared ledgers müssen die verschiedenen Akteure eines Produktentwicklungsprozesses nicht mehr durch einen Vermittlungsmann gehen um Informationen und Updates für ein spezifisches Produkt zu bekommen. Alle Akteure teilen sich den gleichen Ledger und können dadurch jederzeit die Daten abfragen und neue Statuseinträge über jedes Produkt bereitstellen. Somit würde Blockchain in diesem möglichen Use Case eine bessere Interaktion und Transparenz vom Produkthersteller bis hin zu dem Lieferanten und, schlussendlich, dem Kunden ermöglichen.

#### **5. Artificial Intelligence (AI)**

AI bewegt sich aktuell sehr schnell von der Forschungsabteilung hin in die Produktionshalle. Dabei stehen DNN und Deep Learning im Mittelpunkt der Künstlichen Intelligenz (KI) (engl. Artificial Intelligence). Grund dafür ist das breite Spektrum von Anwendungsmöglichkeiten, welches DNN und Deep Learning bieten. Von der Objektidentifikation – ein kritischer Aspekt für Anwendungen wie die Fahrerassistenzsysteme für Kraftfahrzeuge (ADAS) – bis hin zur Handschrift- und Spracherkennung: AI und seine Teilbereiche DNN und Deep Learning sollten Embedded-Fachleute auch 2018 unbedingt am Radar behalten. Ein weiteres Beispiel zu AI finden Sie auch in unserem [Zühlke Blog](#).