

Embedded Realtime on ARM uC - FreeRTOS auf STM32F4

Michael Ebert
Rotdornweg 28
Ad-hoc Networks GmbH
25451 Quickborn
ebert@ad-hoc.network

Christoph Bläßer
christph.blaesser@gmx.de

KURZFASSUNG

Die Geschichte der Gummibärchen ist voller Überraschungen. . .

Stichwörter

RTOS, ARM , STM32, Real Time.

In dieser Abhandlung wurde die Geschichte neu interpretiert. Es ergaben sich völlig neuartige Forschungsansätze, die so vielfältig sind, dass sich die Auswirkungen gegenwärtig kaum abschätzen lassen. Auch Artikel mit vielen Autoren [2] befassen sich mit diesem Thema.

1. EINLEITUNG

Im Rahmen des vorliegenden Papers wird das Echtzeitbetriebssystem FreeRTos vorgestellt. Hierzu werden zu Beginn die allgemeinen Vorgaben für Echtzeitbetriebssysteme beschrieben. Im Verlauf des Textes wird an ausgewählten Beispielen dargestellt, wie freeRTos diese Anforderungen berücksichtigt und durch geeignete Programmfunktionen umsetzt. [1, 3].

2. ÜBERSICHT UND GRÜNDE FÜR DEN EINSATZ EINES RTOS

Mit der steigenden Leistungsfähigkeit von modernen uProcessoren, steigen auch die Anforderungen an die Software die auf diese Systeme aufsetzt. Viele dieser Anwendungen verlangen trotz ihrer Komplexität, dass der Programmlauf der Software oder zumindest einige Teile dieser Software in bestimmten zeitlichen Grenzen ausgeführt wird und somit vorhersehbar und deterministisch ist. Ein Echtzeitbetriebssystem (RTOS) bietet einem Entwickler die Möglichkeit ein solches System zu entwerfen. Determinismus ist das Kennzeichen jedes Echtzeitbetriebssystem, für viele Anwendungen bietet ein Echtzeitbetriebssystem aber noch andere Vorzüge.

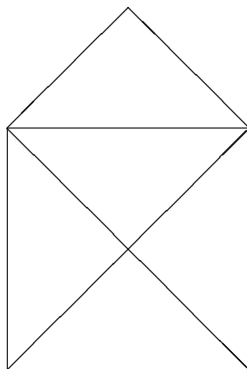


Abbildung 1. Das Haus des Nikolaus in seiner ersten, ursprünglichen Form, siehe auch [3, S. 93].

Und in Tabelle 1 ist das Ganze tabellarisch dargestellt.

3. ZUSAMMENFASSUNG

Literatur

1. http://www.acm.org/class/how_to_use.html.
2. M. Y. Ivory and M. A. Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Comput. Surv.*, 33(4):470–516, 2001.

Literatur

1. How to classify works using ACM's computing classification system. http://www.acm.org/class/how_to_use.html.
2. D. L. Black, D. B. Golub, K. Hauth, A. Tevanian, and R. Sanzi. The Mach exception handling facility. In *Proceedings of the 1988 ACM SIGPLAN and SIGOPS workshop on Parallel and distributed debugging*, PADD '88, pages 45–56, 1988.
3. M. Y. Ivory and M. A. Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys*, 33(4):470–516, 2001.

| Material | Tierart | essbar |
|----------|---------|--------|
| Gummi | Bär | ja |

Tabelle 1. Eine Übersicht zu den Fachbegriffen.