32 Bit µC for Embedded Design

ARM "ARM7"

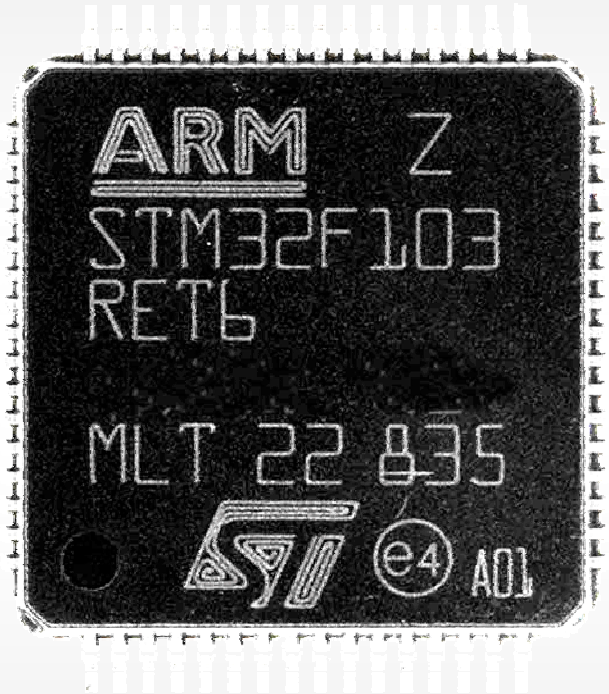
- > 48 000 Gates
- 0.95 Dhrystones
- > 0.06 mW/MHz

Motorola/Freescale "Coldfire"

- > 46 000 Gates
- 1.05 Dhrystones
- > 0.05 mW/MHz

ARM "CortexM3" (Luminary, NXP, ST)

- > 33 000 Gates
- 1.25 Dhrystones
- > 0.047 mW/MHz
- Lots of periphery on chip



Mobphones: ARM-Marktanteil 90%

CortexM3 Embedded Control (8 Bit) -> "**Embedded Numerics (32 Bit)**"

Kennzeichen

- 32 Bit Harvard-Architektur (getrennte Daten- und Befehlsbusse)
- Aber linearer, gemeinsamer Adressraum (4 Giga) für ADR, DATA, IO
- ARM7-Nachfolge auf kleinen 16-Bit Befehlssatz "Thumb2" reduziert
- Multiplikation + Division von 32 Bit in einem Zyklus
- 30…100 MHz, 36 Pin … 144 Pin, typ. 40mA/5V
- Flash bis 512 kB on bord, RAM bis 512 kB
- Flash-Loader über UART1 vorprogrammiert (STM32 F103 RET6)
- Debug über JTAG oder Serial Wire Debug (SWD)
- Drei-Register-Befehle: ADD r0, r1, r2 ; // r0 := r1 + r2;
- CMSIS (Cortex Microcontroller Software Interface Standard - ARM)
- 3…5x UART, SPI, I²C, (Eth), (USB), PWM, Timer, NVIC, ADC, DAC, (SSI)

Allg. Infos

- http://infocenter.arm.com
- http://www.mikrocontroller.net/articles/ARM
- http://de.wikipedia.org/wiki/ARM_Cortex-M3

Vergleich ARM7 <-> CortexM3

Figure 1. Relative performance for ARM7TDMI-S (ARM) and Cortex-M3 (Thumb-2)

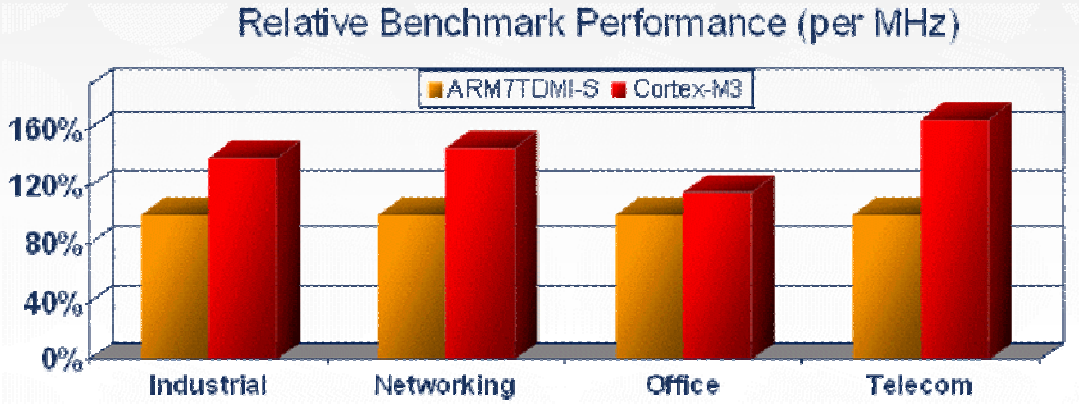
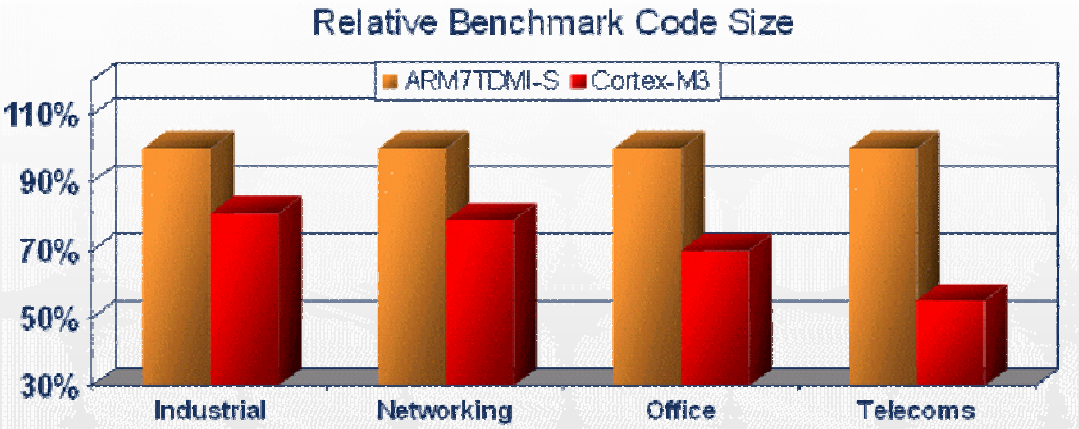


Figure 2. Relative code size for ARM7TDMI-S (ARM) and Cortex-M3 (Thumb-2)

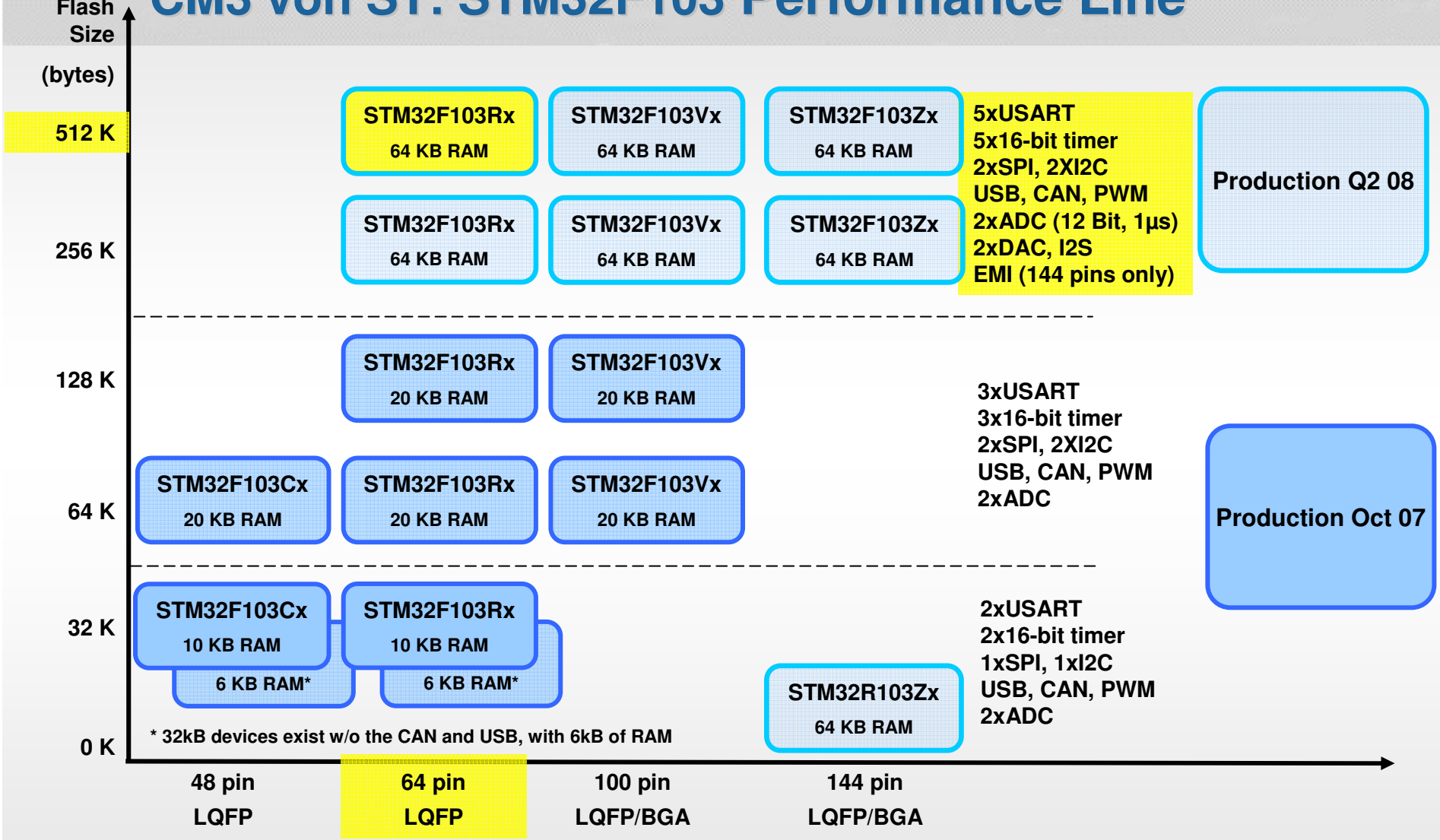


CortexM3

Hersteller

- Luminary Micro (jetzt Texas Instr.), CortexM3-Pionier
  Stellaris-Familie LM3Sxxxx, 72MHz, ADC 10 Bit, incl. PHY
  http://www.luminarymicro.com/products/products.html
- NXP (Philips) Serie LPC1xxx, 70 MHz, ADC 12Bit, 512kB
  http://www.standardics.nxp.com/products/mcus/cortex-m3/
- SGS-Thomson (ST) Serie STM32F10x, 100 MHz, 512kB, ADC 12Bit/1µs
  http://www.st.com/mcu/inchtml-pages-stm32.html
- Toshiba Serie TMPM330, 33 MHz, 256 kB
  http://www.toshiba-components.com/microcontroller/TMPM330.html
- Atmel Serie AT91 / SAM3U, 2x128 kB
  http://www.atmel.com/products/at91/sam3landing.asp?family_id=605
- Energymicro (EFM) 32 MHz, ultra low power, ab 2010
  http://www.energymicro.com

CM3 von ST: STM32F103 Performance Line



- Siehe auch http://www.st.com/mcu/familiesdocs-110.html

Software Tools

Kommerzielle Tools (Einzelplatzlizenz 1000-3000 €), kein Lib.-Sourcecode

- Keil https://www.keil.com/update/sw/rvmdk/3.50
- IAR http://www.iar.com/website1/1.0.1.0/68/1/
- Crossworks http://www.rowley.co.uk/
- Codesourcery http://www.codesourcery.com (G++)

Verdeckt kommerziell/ teilkommerziell

- Raisonance http://www.raisonance.com/ (Rlink) (+install!) (900,-€)
- EMEDT/ CooCox-Colink http://www.emedt.com/armdt/ (JTAG)

Open Source

- Amontec (JTAG) www.amontec.com/openocd.shtml sdk4arm (yagarto)
- Olimex (JTAG) www.olimex.com ("incl. CD for Dummies")
- YAGARTO M. Fischer (ohne install-script) http://www.yagarto.de/
- WinARM (siwawi) M. Thomas (ohne install-script) http://www.siwawi.arubi.uni-kl.de/avr_projects/arm_projects/

Software Toolchain STM32F103

http://www.st.com/mcu/familiesdocs-110.html

- GNU-C Compiler (latest: GCC 4.4.1)*
- Linker Scripts
- GNU BinaryUtilities
- GNU C/CPP-Compiler
- GNU-Debugger
- C-Libraries, Runtime-Libs
- IDE (Eclipse o.a.)
- Simulator
- JTAG-Interface (möglichst openOCD)

* (Thumb2-Compiler ist von Codesourcery)

Cortex Microcontroller Software Interface Standard (CMSIS)

- for Cortex-M0 / Cortex-M1 / Cortex-M3 processors, Future: Cortex-Mx
- various silicon and software vendors
- interface to peripherals,
- real-time operating systems,
- middleware components.

Software layers available for various compiler implementations:

- **Core Peripheral Access Layer**: interface for RTOS Kernels that includes debug channel definitions.
- **Middleware Access Layer**: device specific peripherals used by middleware components.
- **Device Peripheral Access Layer**: provides definitions for all device peripherals
- **Access Functions for Peripherals** (optional): helper functions for peripherals

CMSIS defines:

- names/methods: peripheral registers, exception vectors, core peripherals
- device independent interface for RTOS Kernels including debug channel
- Interfaces for middleware components (TCP/IP Stack, Flash File System)

Software Toolchain

Wovon reden wir?

Typische Builts bestehen aus

- 100-170 MB
- 3000-5000 Files (!) davon 300 an das Problem anzupassen
- "kein Problem, machen wir mal schnell?"

Prozessor-spezifische Anpassung in hunderten Config-, Header- und C-Files braucht Monate: für kmU u.U. tödlich -> Hände weg von opensource!

Erfolglose Installationen an der GFaI (6 Monate vergeudet):

- Keil µvision3 -> Test mit Stellaris LM3S811 Evalkit, Prozessor fehlte
- YAGARTO (M. Fischer) -> thumb2-Compiler fehlte, ohne Install
- Crossworks -> 30 Tage genügten nicht, um Compiler zu erwecken
- Amontec sdk4arm -> drei Tools ohne Beschreibung
- WinARM (M. Thomas) -> ohne Install, Doku unzureichend
- armDT (EMEDT) -> spezielles Debug-Interface, Doku unzureichend

Raisonance Ride7

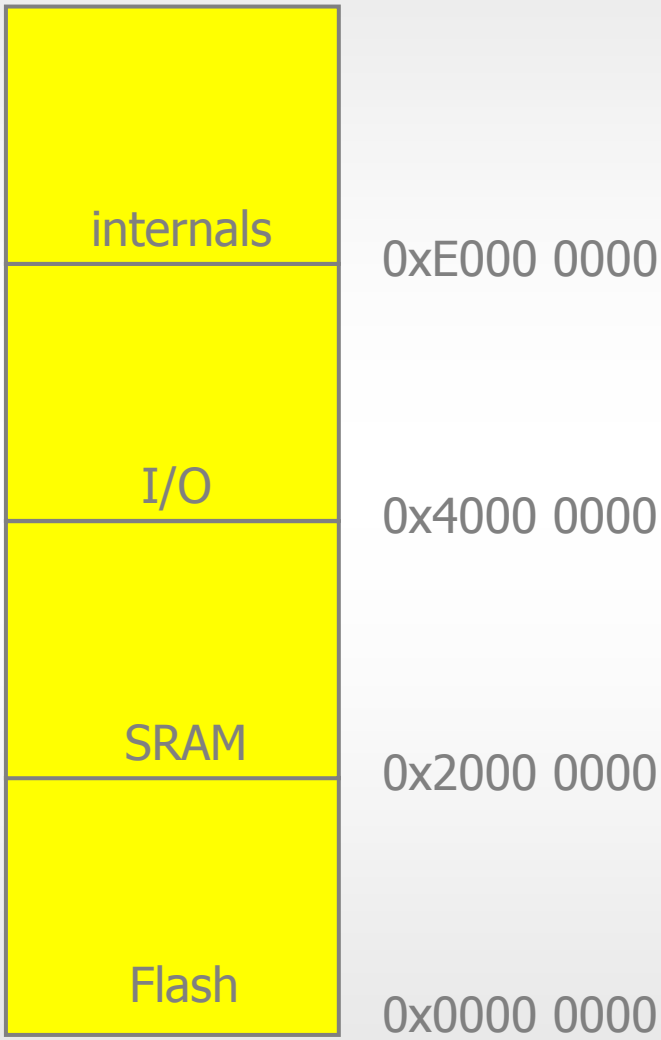
- Rund 100 Beispielprogramme STM32F103 (10/2009) downloadbar unter http://www.stm32circle.com/projects/list.php
- (incl. USB & Ethernet Beispiele)
- Installiert sich komplett selbst
- Helps sind dabei (wenige)
- Derzeit beste Akzeptanz bei STM32F103-Projekten
- Anwender-Kommentar:
  "Auf der Page von Raisonance.com anmelden und die Files "RKit-ARM" (GCC) und "Ride7" (IDE) downloaden und installieren. Zuerst RKit-ARM dann Ride7. Der Umgang mit diesen Programmen ist super einfach - keine Rumquälerei mit Makefiles und solcherlei Teufelszeug - alles ist bereits Mundgerecht von Raisonance zugeschneidert. Sehr empfehlenswert!"

Wenn und Aber:

- Codelimit 32kB, Vollversion 900 € Einzelplatz

Adressraum

Adressierbar:
32 Bit ~ 0xffff ffff = 4.3 Giga

STM32F103RET6

- 64k SRAM
- 512k Flash (EEPROM)