Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Inter-MCU-Platform Hardware Analysis Towards a Clean-slate Timer-API for RIOT-OS

INET Seminar / MINF-GSM

Niels Gandraß <Niels.Gandrass@haw-hamburg.de>
Michel Rottleuthner <Michel.Rottleuthner@haw-hamburg.de>

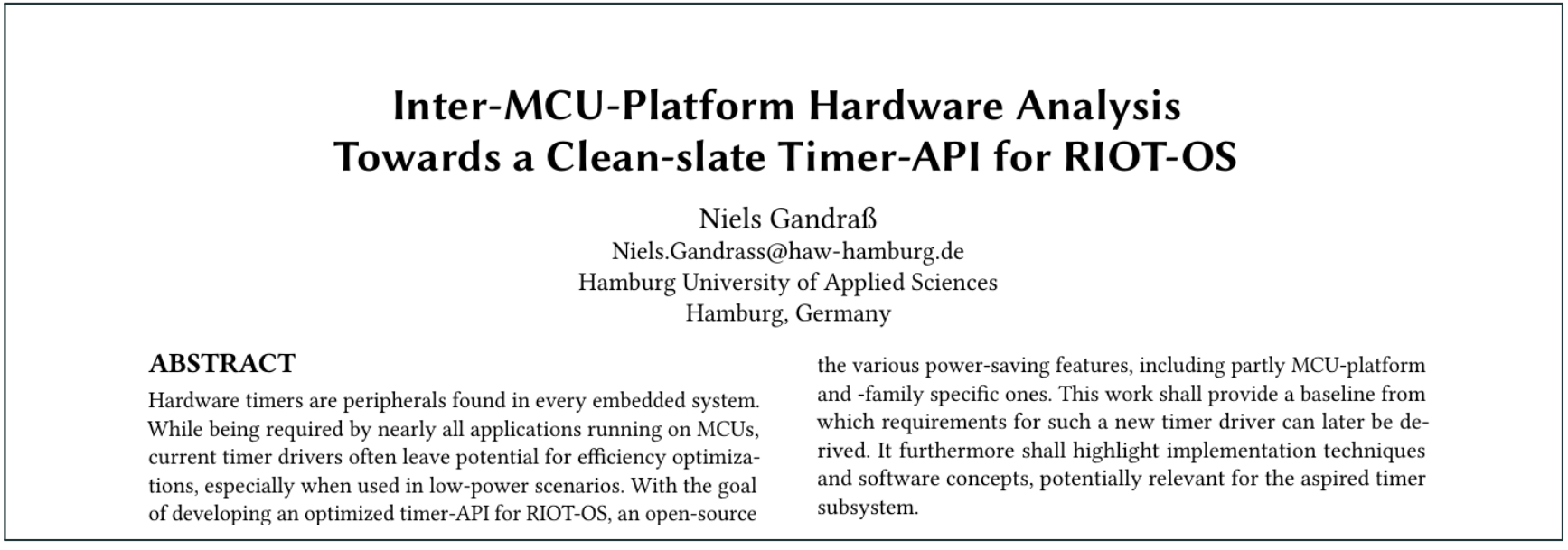
February 11th, 2020

Hamburg University of Applied Sciences
Faculty of Engineering & Computer Science

Table of Contents

Introduction

Introduction



Inter-MCU-Platform Hardware Analysis Towards a Clean-slate Timer-API for RIOT-OS

Niels Gandraß
Niels.Gandrass@haw-hamburg.de
Hamburg University of Applied Sciences
Hamburg, Germany

ABSTRACT

Hardware timers are peripherals found in every embedded system. While being required by nearly all applications running on MCUs, current timer drivers often leave potential for efficiency optimizations, especially when used in low-power scenarios. With the goal of developing an optimized timer-API for RIOT-OS, an open-source the various power-saving features, including partly MCU-platform and -family specific ones. This work shall provide a baseline from which requirements for such a new timer driver can later be derived. It furthermore shall highlight implementation techniques and software concepts, potentially relevant for the aspired timer subsystem.

Figure 1: Corresponding paper containing all details outlined in this talk

Introduction

Our contribution is twofold:

Review of Related Work

Research addressing timers from both a hardware as well as a software point-of-view is depict.

Analysis of Timer Hardware

Timer peripherals of MCU-platforms, currently supported by RIOT-OS, are compared.

Long-term Goal: New Clean-slate Timer Subsystem for RIOT-OS

Backed on the results of a comprehensive analysis of existing timer hardware.

Introduction

TL;DR: I'll talk about how we read a whole lot of MCU documentation...

Figure 2: Quickly print the whole Wikipedia - what if? #59

Conferences & Journals

Major Conferences

- USENIX Annual Technical Conference[1] (USENIX ATC)
- USENIX Symposium on Operating Systems Design and Implementation[2] (USENIX OSDI)
- ACM Symposium on Operating Systems Principles[3] (SOSP)
- ACM Conference on Embedded Networked Sensor Systems[4] (SenSys)

---

[1] USENIX ATC website archive: `https://www.usenix.org/conferences/byname/131` (Accessed 13.01.2020)

[2] USENIX OSDI website archive: `https://www.usenix.org/conferences/byname/179` (Accessed 30.01.2020)

[3] ACM SOSP website: `http://www.sosp.org/` (Accessed 30.01.2020)

[4] ACM SenSys website: `https://sensys.acm.org/` (Accessed 13.01.2020)

Major Conferences (Continued)

- ACM/IEEE International Conference on Information Processing in Sensor Networks[5] (IPSN)
- International Conference on Embedded Wireless Systems and Networks[6] (EWSN)

[5] IPSN website: `https://ipsn.acm.org/` (Accessed 13.01.2020)
[6] EWSN website: `http://www.ewsn.org/` (Accessed 13.01.2020)

Specialized Conferences

[7]

[7] RIOT Summit Website: `https://summit.riot-os.org/` (Accessed 13.01.2020)

Publication Sources - Online Libraries

[8]

[9]

[10]

[8] IEEE Xplore website: `https://ieeexplore.ieee.org/` (Accessed 13.01.2020)
[9] ACM Digital Library website: `https://dl.acm.org/` (Accessed 13.01.2020)
[10] unreviewd pre-print only, ArXiv website: `https://arxiv.org/` (Accessed 13.01.2020)

Publication Sources - Journals and Letters

- ACM Transactions on Sensor Networks[11] (TOSN)
- ACM Transactions on Embedded Computing Systems[12] (TECS)
- ACM SIGOPS Operating Systems Review[13] (OSR)
- IEEE Internet of Things Journal[14] (IEEE IoT)

[11] TOSN in the ACM-DL: `https://dl.acm.org/journal/tosn` (Accessed 30.01.2020)
[12] TECS in the ACM-DL: `https://dl.acm.org/journal/tecs` (Accessed 30.01.2020)
[13] OSR in the ACM-DL: `https://dl.acm.org/newsletter/sigops` (Accessed 30.01.2020)
[14] IEEE IoT on IEEE Xplore: `https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=6488907` (Accessed 30.01.2020)

Publication Sources - Academic Search Engines

Google Scholar [15]

Microsoft Academic [16]

Semantic Scholar [17]

[15] Google Scholar website: `https://scholar.google.com/` (Accessed 27.01.2020)
[16] Microsoft Academic website: `https://academic.microsoft.com/` (Accessed 27.01.2020)
[17] Semantic Scholar website: `https://www.semanticscholar.org/` (Accessed 27.01.2020)

Related Work

Related Work

Publications were selected according to the anticipated relevance for our work, as estimated to the best of our knowledge.

Highlighted research was split into two categories:

Timer Hardware - Timers from an hardware point-of-view

- Generic properties and functions of timer peripherals
- Comparisons of different MCU-platforms or -families

Software Modules - Timer drivers and usage in timer subsystems

- Generic design aspects and implementation concepts
- Application-specific timer implementations
- Real-time scheduling based approaches

Related Work - Overview

Timer Hardware

2x Generic properties and functions of timer peripherals

- [Ka11], [SM05, pp. 67-68, pp. 87-89]

2x Comparisons of different MCU-platforms or -families

- [MS05, pp. 69-91], [Ts14]

Software Modules

6x Generic design aspects and implementation concepts

- [VL97], [MM98], [AD00], [TEF05], [Ts07], [Li16]

8x Application-specific timer implementations

- [GN06], [Be09], [Ps07], [PVB17], [Ba18], [Ha05], [Gr08], [Li16]

4x Real-time scheduling based approaches

- [JK05], [Ho14], [HLS11], [Ho12]

Related Work - Implications for the Hardware Analysis

Timer peripheral properties and functions, as depict in the related work, yield a variety of characteristics for our timer hardware analysis:

Basic Hardware Properties	[Ka11; SM05]
timer type, counter register width, prescaler availability, ...	

Advanced Timer Characteristics	e.g. [Li16] and others
interrupt capability, auto-reload, compare channels, max. resolution, ...	

Power-saving Considerations	[AD00; Be09; Ha05]
low-power clock support, interrupt handling, ...	

Hardware Platform Analysis

Hardware Platform Analysis

We contribute an in-depth analysis of various timer peripherals available in MCUs, currently supported by RIOT-OS.

Key aspects

- A total of 13 MCU-platforms/-families analyzed
- Devices from 8 different manufactures
- Detailed results for every platform
- Inter-MCU-platform timer peripheral comparison
- Shall provide a baseline to derive timer-API requirements from

Hardware Platform Analysis - Scope

The following MCU families were analyzed:

- STMicroelectronics (ST)
  - STM32F0 / F1 / F2 / F3 / F4 / F7
  - STM32L0 / L1 / L2
- Microchip / Atmel
  - ATmega AVR
  - PIC32MX / PIC32MZ
  - SAMD21
- Espressif
  - ESP8266
  - ESP32
- Silicon Labs
  - EFM32 / EFR32
  - EZR32
- Texas Instruments (TI)
  - LM4F120
  - MSP430x1xx / MSP430x2xx
- NXP Semiconductors
  - LPC176x / LPC175x
- Nordic Semiconductor
  - nRF51x / nRF52x
- SiFive
  - FE310-Gx

Hardware Platform Analysis - Methodology

1. Platform selection and information acquisition
   - Based on currently supported MCUs (`/cpu`)
2. Definition of analysis criteria
   - As found in related work and expected promising by us
3. Extraction of timer peripheral details
   - Mind-map format
   - Including all found timer data, even beyond defined criteria
4. Results consolidation into *Timer Comparison Matrices* (TCMs)
   - Overview of all timer types and their characteristics for each platform
5. Deriving of inter-MCU-platform findings
   - Aggregation of TCM results, therefore across all platforms

Hardware Platform Analysis - Extracted Data

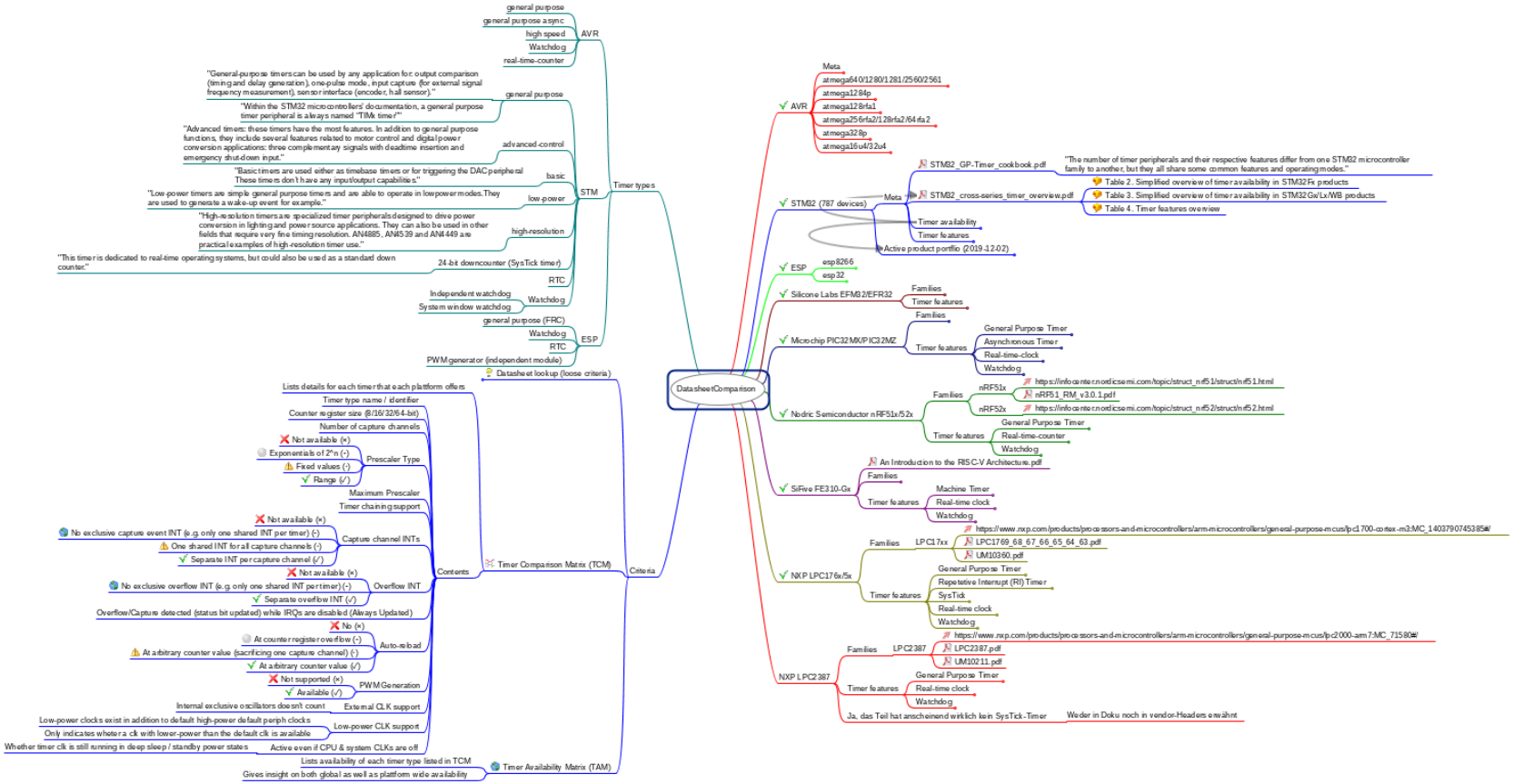


Figure 3: Mind-map containing data for most of the platforms

Hardware Platform Analysis - Extracted Data

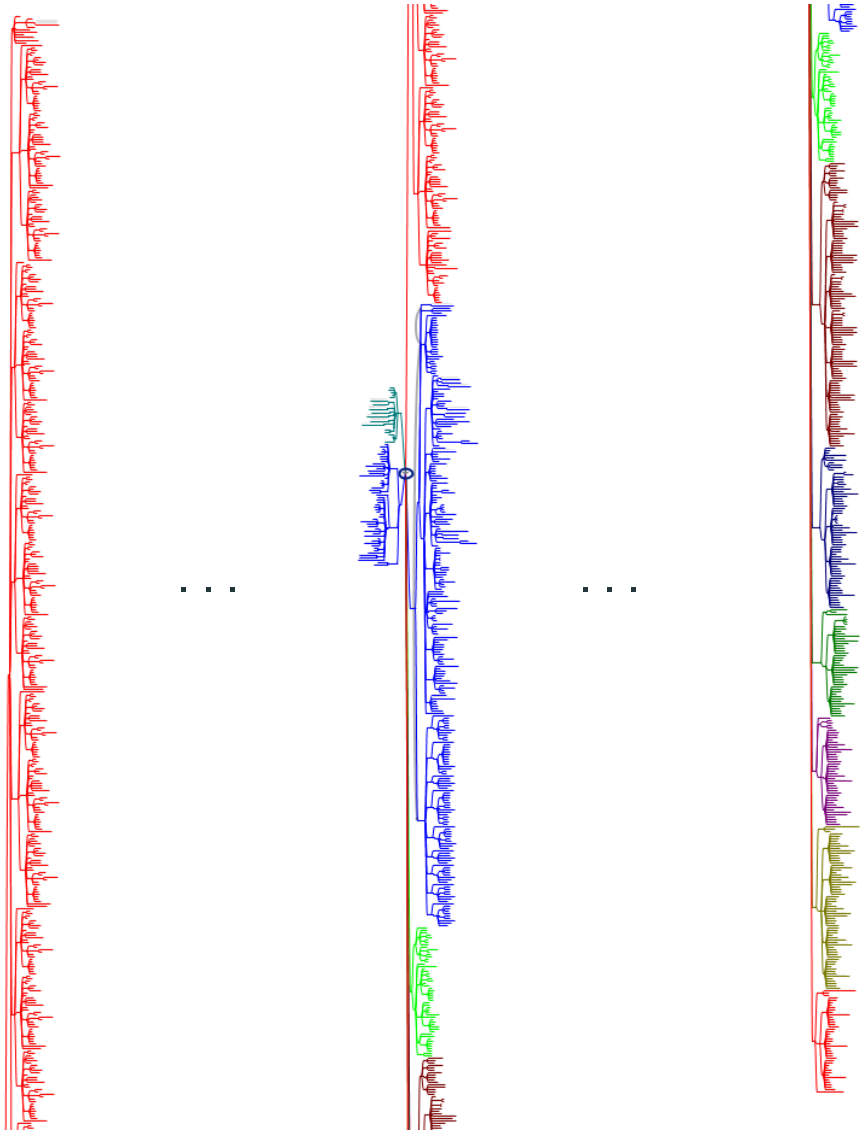


Figure 4: Mind-map containing data for most of the platforms (expanded)

Hardware Platform Analysis - Results

Timer Type	Counter Width	Compare Channels	Prescaler Type	Max. Prescaler	Chaining Support	Compare INT	Overflow INT	Event Flags	Auto-reload	PWM Generation	External CLK	Low-power CLK	Deep-sleep Active
General-purpose	16 bit 32 bit	1-4	R	2^{16}	✓	○	○	✓	✓	✓	✓	×	×
Advanced-control	16 bit	4, 6	R	2^{16}	✓	○	○	✓	✓	✓	✓	×	×
Basic	16 bit	0	R	2^{16}	✓	×	○	✓	✓	×	×	×	×
Low-power	16 bit	1	E	2^{7}	×	○	×	✓	✓	✓	✓	✓	×
SysTick	24 bit	0	F	2^{3}	×	×	✓[b]	✓	✓	×	×	×	×
Real-time-clock	-	1-2[c]	R[e]	2^{7+15}	×	○	○[d]	✓	-	×	×	✓	✓
Independent WDG	12 bit	0	E	2^{8}	×	×	×	-	×	×	×	✓[e]	✓
System window WDG	7 bit	0	E	2^{12+3}	×	×	×	-	×	×	×	×	×

Table 1: Timer Comparison Matrix: STMicroelectronics STM32

Hardware Platform Analysis - Results

ID	Title	Description	Criterion	Platform	Timer	Platform	Timer
R-01	Counter width	Usable size of the counter register in bits (Excluding watchdog timers)	≥ 16	13	49	100 %	85 %
			≥ 32	11	21	85 %	36 %
			≥ 64	3	3	34 %	5 %
R-02	Compare channels	Number of available compare channels (Excluding timers w/o compare channels)	≥ 1	13	50	100 %	100 %
			≥ 2	9	32	69 %	64 %
			≥ 4	7	11	54 %	22 %
R-03	Prescaler	Support for prescaling the timer clock	*yes*	13	53	100 %	75 %
R-04	Timer chaining	Support for timer module combination (Excluding watchdogs and RTCs)	R-01 ≤ 16	6	9	67 %	38 %
			R-01 > 16	3	5	23 %	24 %
R-05	Compare interrupts	Unique INTs for each compare channel	*yes*	6	17	46 %	30 %
R-06	Overflow interrupts	Unique INTs for counter over-/underflow (Excluding watchdogs)	*yes*	4	8	31 %	20 %
R-07	Event flags	Availability of status bits for timer events	*yes*	10	60	100 %	100 %
R-08	Auto-reload	Auto-reload at over/-underflow (OVF), at compare-channel match (CCM), or via auto-reload register (ARR) (Excluding watchdogs and RTCs)	OVF	3	9	23 %	17 %
			CCM	4	17	31 %	32 %
			ARR	6	27	46 %	51 %
			any	13	53	100 %	100 %
R-09	Low-power clock	Low-power oscillator can be used by timer	*yes*	13	51	100 %	70 %
R-10	Deep-sleep active	Timer operational in lowest MCU power states	*yes*	13	45	100 %	62 %
R-11	GP-timers	Number of available general-purpose timers	$= 1$	1	-	8 %	-
			≥ 1	12	-	92 %	-
R-12	WDT interrupts	Watchdog generates interrupt prior to reset	*yes*	9	10	69 %	67 %
R-13	Unknown items	Timer has unresolved/unknown properties	*yes*	7	14	54 %	19 %

Table 2: Selected overall results across all 13 analyzed MCU platforms

Hardware Platform Analysis - Results

Exemplary excerpt from our results:

Counter range [R-01, R-03, R-04]

High counter range desirable: lengthens time between over-/underflows, therefore reduces maintenance overhead and wakeups

- All MCUs provide at least 16-bit, 85 % even 32-bit, wide counters
- Platforms offering only $\leq$ 16-bit timers often support timer chaining for counter width extension
- Prescalers available for 75 % of all timers
  - Keep resolution / max-timeout trade-off in mind
- The only non-prescalable general-purpose timer is 64-bit wide

See paper Section 4.3.1 for details

Hardware Platform Analysis - Results

Exemplary excerpt from our results:

Low-power operation and energy saving [R-09, R-10]

Power-saving features highly desired: entering MCU sleep states, keeping timer running while CPU and HF-peripheral clock are powered down

- 70 % of all timers can use a low-power clock
- Every platform offers basic low-power timer peripherals
- Specialized *always-on* timers are also provided by every platform
  - Timers that are able to operate in even the lowest power states
- These features must be properly utilized by a timer subsystem

See paper Section 4.3.5 for details

Hardware Platform Analysis - Outstanding Tasks & Issues

The conducted analysis already yielded insight into a broad range of properties and features. However, various outstanding tasks remain:

- Analysis of remaining platforms and resolving of open questions
- Closer examination of ...
  - Clock-tree and available oscillators
  - Peripheral interconnect buses
  - Event triggered hardware tasks
  - Configuration and maintenance costs

Outlook & Future Work

Future Work

After completing outstanding tasks from the hardware platform analysis, we picture the following future work:

Next steps

1. Definition of abstract timer classes
   - Platform agnostic feature-sets that allow classification of timers
2. Conducting a hardware availability analysis
   - According to defined timer classes
   - Allows impact estimation of different design considerations
3. Deriving requirements for the aspired RIOT-OS timer subsystem
   - Uniting hardware analysis results and techniques from related work
   - With respect to different usage scenarios and characteristics

Future Work

There still is a lot of paperwork that needs to be done...

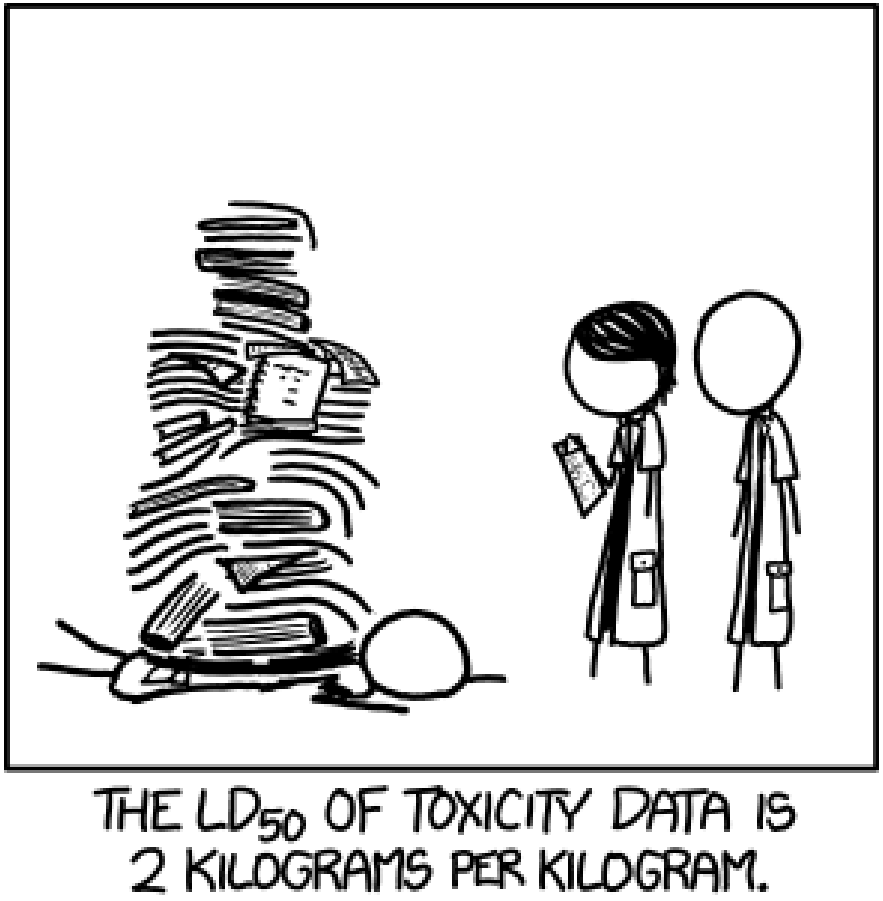


Figure 5: LD50 - xkcd #1260

Questions?

Discussion!