

Advanced Embedded Development in Linux

OpenAlt 2017

Josef Kubín

Introduction

- **Part 1 (tracing)**
 - What is a real time system?
 - Available tracing tools
 - The most trivial tracing
 - How to trace USB, Ethernet
 - Trace log processing by Vim
 - I have to understand a terrible C source ...
- **Part 2 (scripting)**
 - Windows compilers in Linux environment
 - (make) Scripts

What is a real time system?



(Comedy Juggler James BuStar)

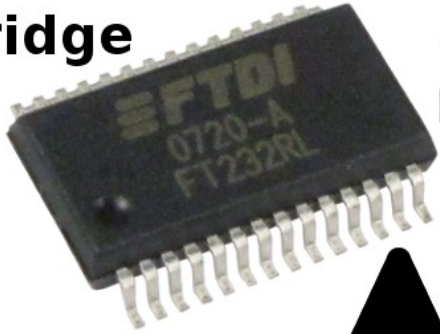
Professional tracing tools

(not so cheap proprietary closed black boxes)

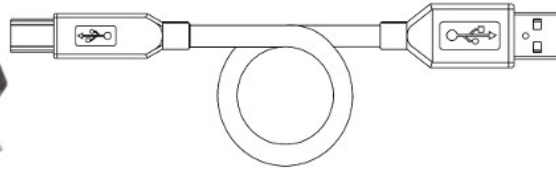


The most trivial tracing

Bridge



USB Full Speed (12Mb/s)



UART RX

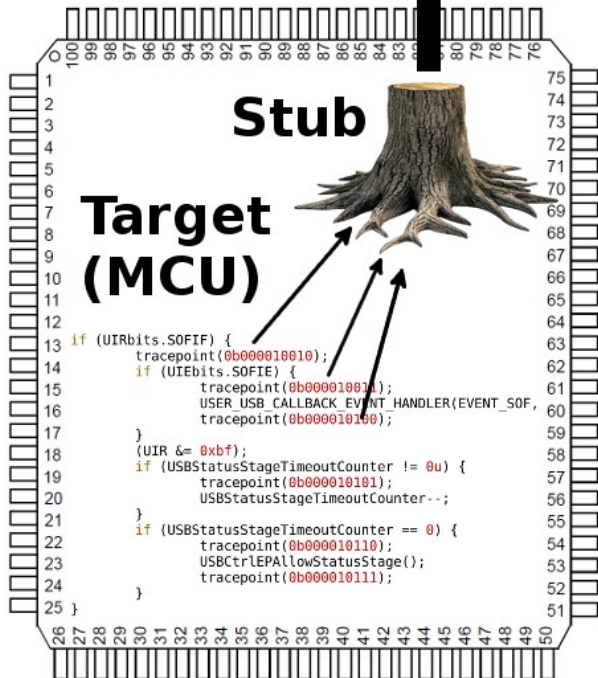
**Trace stream
(max. 3MBd)**

UART TX



Stub

**Target
(MCU)**



Basic tracing terms

- **Bridge**
usually an FTDI chip between PC and an examined circuit (MCU)
- **FTDI (Future Technology Devices International)**
chip maker of USB bridges, commonly known by its abbreviation
- **tracepoint(ID);**
a tracking point in the source code that sends the tag (ID) to a stub
- **Stub**
a small piece of software (HW dependent) inside of a target which sends trace data to an external interface (usually UART)
- **Target**
the examined circuit, usually a microcontroller (MCU)
- **TSM (Trace Synchronisation Mark)**
time or data stream sync sign for an external tracing software

The most trivial stub (PIC18), max. 256 unique tracepoints



```
#include <p18cxxx.h>
```

```
void traceinit(void)
{
    /* 500000Bd; CPU speed 64MHz (16 MIPS) */
    SPBRG1    = 31;
    BAUDCON1  = 0b00001000;
    TXSTA1    = 0b00100100;
    RCSTA1    = 0b10000000;
}
```

```
void tracepoint(unsigned char id)
{
    /* UART barrier */
    while (!PIR1bits.TX1IF);

    /* write trace ID to UART buffer */
    TXREG1 = id;
}
```


FT2232H 2x bridge with 4kB buffers! 12MBd, USB High Speed (480 Mb/s)

CFSUNBIRD



How do I link tracepoints with an examined source code?

**How do I link tracepoints with an
examined source code?**

ctags!

Vim shortcuts for trace.log

1) Highlight trace ID (or something)

2) Keep cursor on trace ID and open preview window

Ctrl-h

3) Set vertical windows to equal width

Ctrl-w =

- Line up

k

- Line down

j

Tracing automaton

- Implements fast, tiny and binary independent protocol
 - **Naturally no data for symbol table (ELF/COFF) to connect tracepoints with a source code tree!**
 - Reminds CISC instruction set (1B opcode + payload)
- Tracing automaton recognizes
 - Tracepoints (with timestamps), max. 2^{31} points
 - Signed/Unsigned integer 1/2/4/8 Bytes
 - Bit fields 8/16/24/32 bits
 - All known floating point numbers (2 + 2 from 16)
 - Raw data (with possible RLE 512B compression)
 - Strings
 - 16 structures (mix of previous)

opentracer



<https://github.com/jkubin/opentracer>

an open source tracing tool for small processors

Josef Kubín

What is opentracer?

A software tool (set) for ...

- **Development**

- Specific (breakpoint-less) embedded development
USB, Ethernet, RTOS, device drivers, ...

- **Education**

- Helps to understand obscured source code

- **Bughunting**

- Helps to catch intermittent, rarely seen bugs
- A (huge) log file for post mortem analysis

- **Data collection**

- Reads a data from internal devices

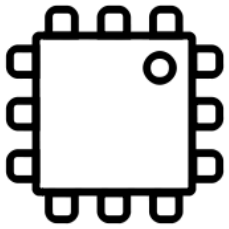
Trace exotic floating numbers

Sign, Exponent, Mantissa

- **minifloat** - just for fun
| seee emmm | or | eeee mmmm |
- **half float** - graphics application
| seee eeemm | mmmm mmmm |
- **Microchip float** - history projects
| eeee eeee | smmm mmmm | mmmm mmmm |
- **extended single (9B)** - summation
| seee eeee | emmm mmmm | mmmm mmmm | mmmm mmmm | mmmm
mmmm | mmmm mmmm | mmmm mmmm | mmmm mmmm | mmmm mmmm |
- **octuple float (16B)** - scientific application
| seee eeee | eeee eeee | eeee mmmm | mmmm mmmm | mmmm
mmmm | mmmm mmmm | mmmm mmmm | mmmm mmmm | mmmm ..
- ...

Join to opentracer project
and try to **write your own stub!**

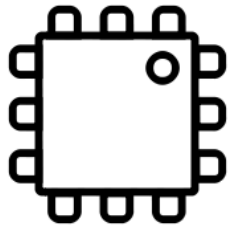
PIC16



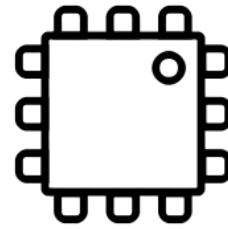
PIC18



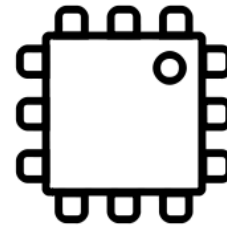
PIC24



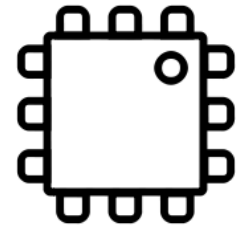
PIC32



AVR



???



<https://github.com/jkubin/opentracer>

opentracer@gmail.com

I have to understand a terrible C source code with ...

... CRLF, Windows paths, ugly coding style, crazy data types, macro misuse, many conditional directives, stale comments, ...

**After weeks of frustration, try to avoid suicide!
Have a rest.**

- 1) Run a **moralizer** script that calls his modest friend named **deodor_ansi**, who does not need an extra header file. He converts crazy data types to generic ansi types.
- 2) Run **gcc** configured as a preprocessor ``gcc -dD -E'` that heals macro madness, straightens conditional directive maze.
- 3) Once you have crystal clear source code by your habits ...
- 4) Put tracepoints, trace data points, **and finally ...**

Happy tracing!



<https://github.com/jkubin/opentracer>

OpenAlt 2017

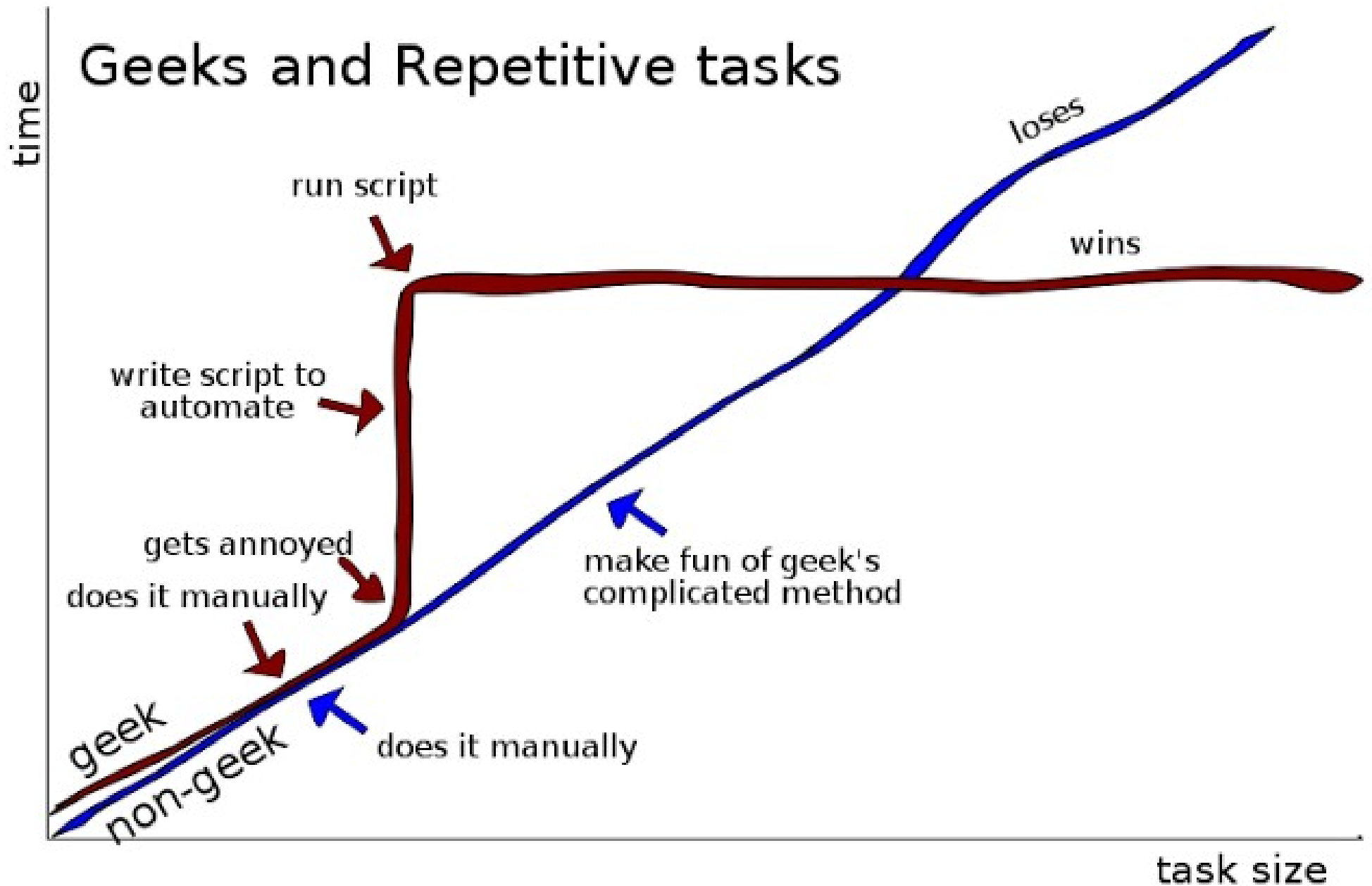
Josef Kubín

Part 2 (scripting)

Those who do not understand UNIX are condemned to reinvent it, poorly.

--Henry Spencer, programmer

(user friendly) IDE versus CLI tools



Děkuji za pozornost!

Nezapomeňte vyplnit anketu!
<http://a.openalt.cz/371>

OpenAlt 2017

Josef Kubín