



Zephyr Project

Industrial & Open-Source Crossover

LLEXT: a new Zephyr subsystem

Presented by: Cedric Lescop

Industrial need: Flexibility



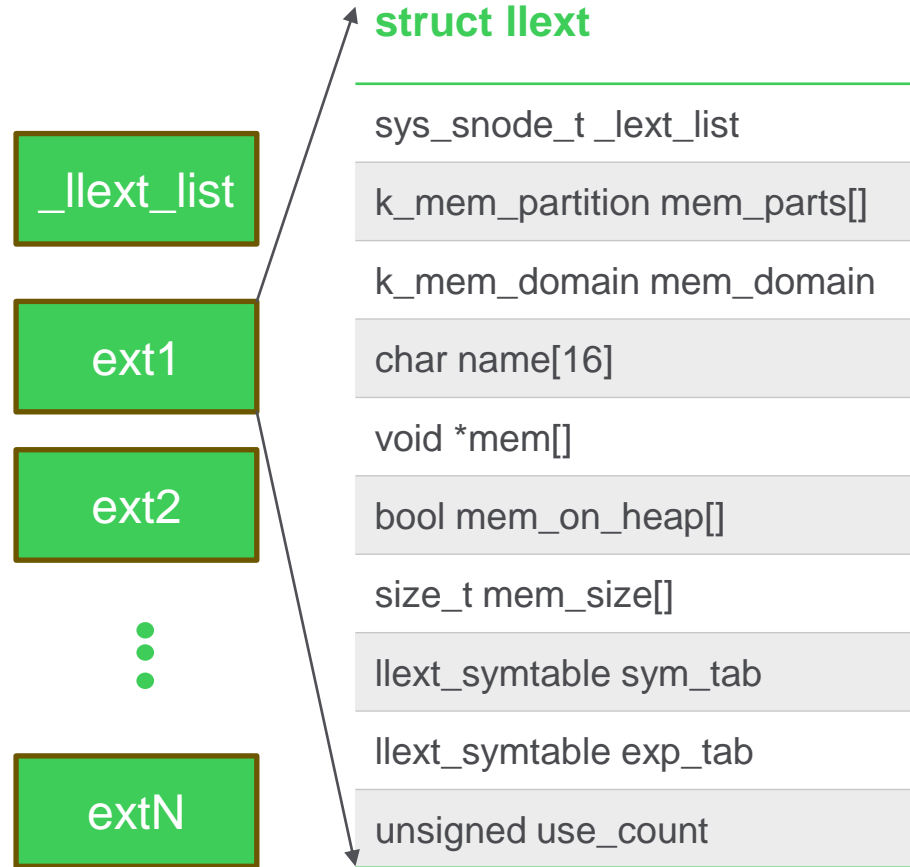
New technology identified: LLEXT subsystem

Linkable Loadable **EXT**ensions state-of-the-art

- What does it do?
 - Load data into memory
 - Using ELF file
- How?
 - Serial back-end
 - Parse the whole instruction code
- Simple test example
 - 1 printk(“Hello, World”)

LLEX: Extension manager

- llex_load(), llex_unload(), llex_find_sym()
- Maintains a list of loaded extensions
- Maintains a reference count for each extension
- Manages memory for extensions
- Manages symbol tables for the base firmware and each extension



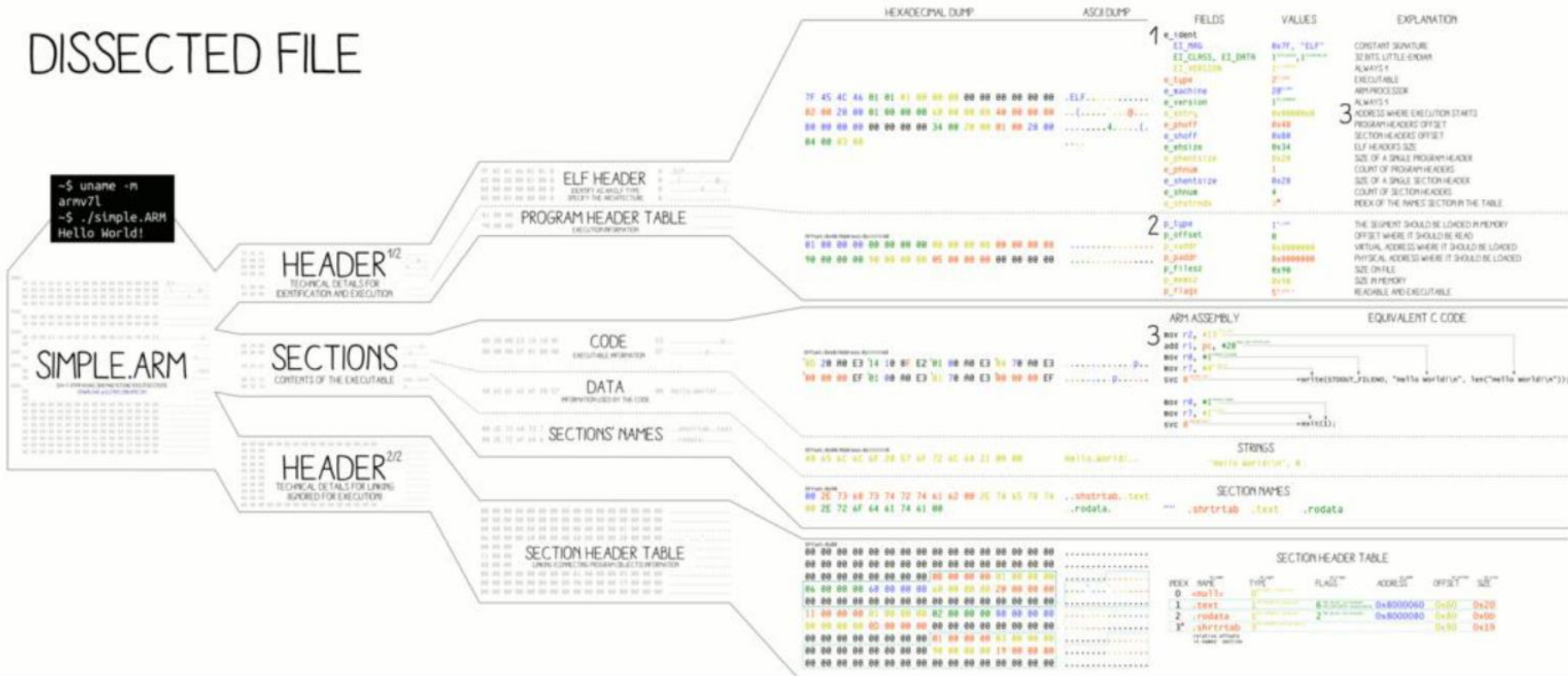
ELF¹⁰¹ a ~~Linux executable~~ walk-through

ANGE ALBERTINI
CORKAMI.COM

Angelo Albertini [CC BY 1.0](https://creativecommons.org/licenses/by/1.0/)

Zephyr Extension

DISSECTED FILE



Relocations: Linking and Placement

Disassembly of section `‘.text’` using `gcc`:

`arm-zephyr-eabi-objdump -r -d -x hello_world.elf` gives:

```
00000000 <hello_world>:  
0: b580      push  {r7, lr}  
2: af00      add   r7, sp, #0  
4: 4b08      ldr   r3, [pc, #32] ; (28 <hello_world+0x28>)
```

```
28: 00000004  .word 0x00000004  
28: R_ARM_ABS32 .rodata
```

```
2c: 00000000  .word 0x00000000  
2c: R_ARM_ABS32 printk
```

```
30: 00000014  .word 0x00000014  
30: R_ARM_ABS32 .rodata
```

```
34: 4718      bx    r3  
36: 46c0      nop                    ; (mov r8, r8)
```

Rewrite with address of `.rodata` + `0x00000004`

Rewrite with address of `printk` function

Relocations: some key points

- Kinds of Relocations that show up depend on
 - Architecture: x86, armv7m, xtensa lx7, etc
 - ELF linkage: shared, static, relocatable
 - Compiler flags: e.g. -mlong-calls
- May require opcode decoding, updating, and reencoding
- May require generation of a jump table
 - Opcodes are sometimes location dependent
 - Limits range of address accessibility for that opcode
 - E.g. PC relative call instructions encoded as 16bit instruction opcode




Impacts on products

Benefits vs. drawbacks

- Updates
 - Partial / isolated / processes
 - Faster application build and load
- Cybersecurity
 - Expose to new threats: untrusted ELF could do damage
- Safety
 - Need new certification?

Credits

LLEXT was created by numerous people, and built on Zephyr infrastructure

- Chen Peng, Initial ELF Loader (Former )
- Lucas Burelli, Armv7m and CMake tooling (Arduino)
- Guennadi Lyakh, Xtensa Support ( - Sound Open Firmware)
- Ederson de Souza, EDK ( - Zephyr Team)

A special thank you to **Tom Burdick** for gluing it together and letting me largely copy his presentation!



Life Is On | Schneider
Electric

se.com

