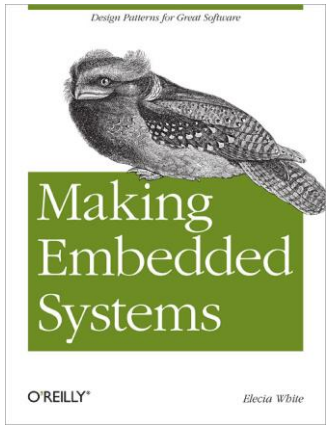
Embedded
Online
Conference

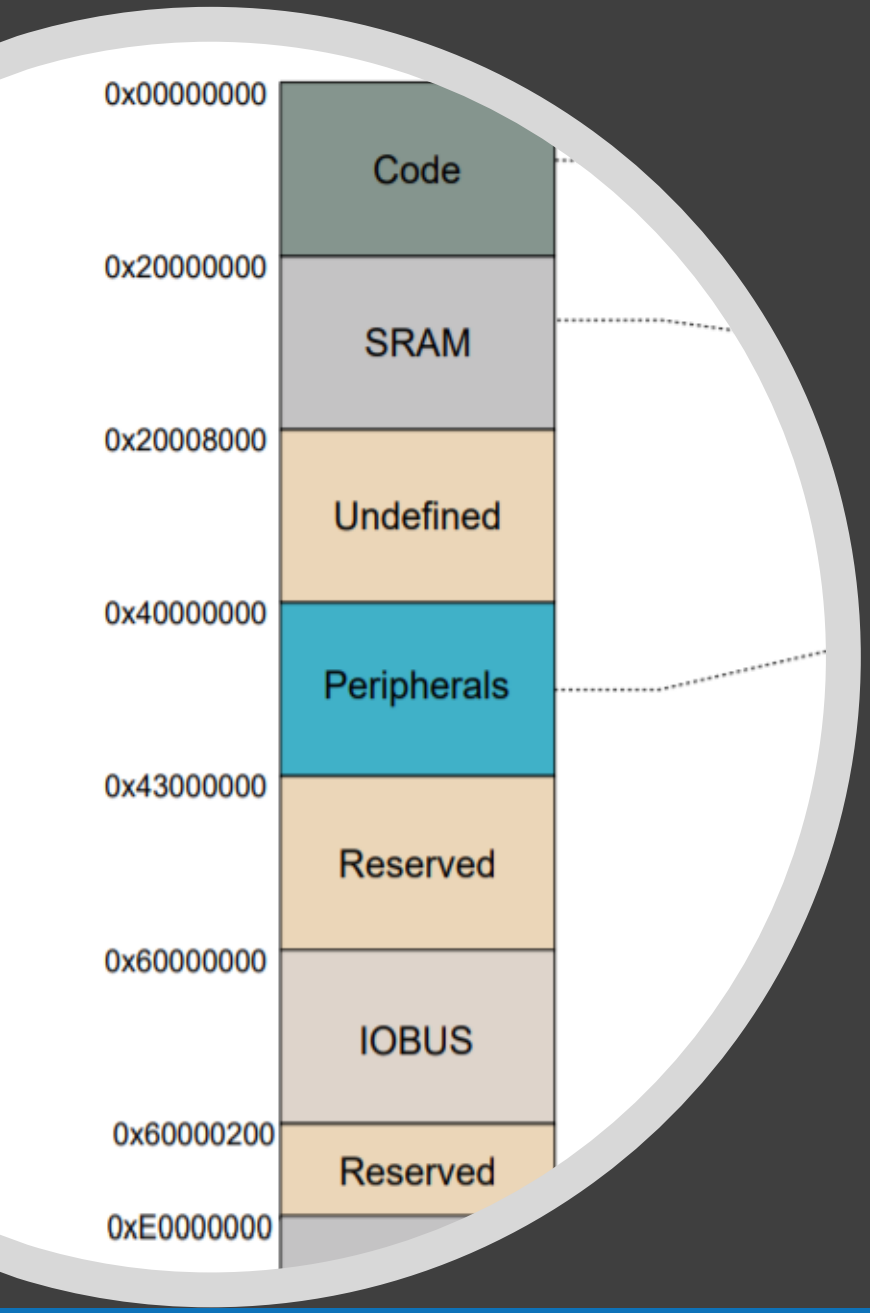


www.embeddedonlineconference.com

Buried Treasure and Map Files

Ahoy there, matey!

Elecia White *@logicalelegance*



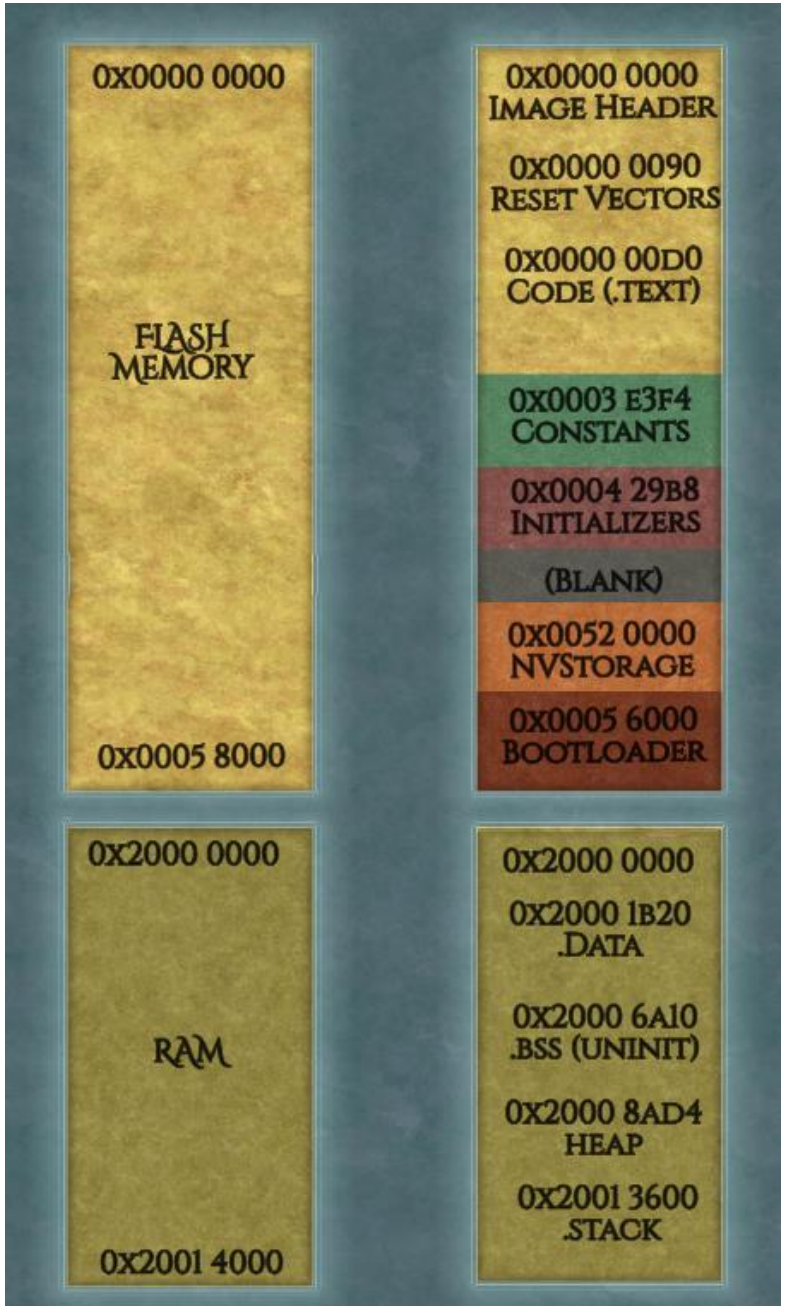
A	B	
	Start	Siz
Flash Memory	0x0000 0000	0x58
RAM	0x2000 0000	0x140
Flash	Start	Size
Image Header	0x0000 0000	0xCF
Code	0x0000 00D0	0x47
NV Storage	0x0005 2000	0x3
otloader	0x0005 6000	0
End	0x0005 8000	

There are many ways of looking at memory

Memory Maps

Memory Layout

Planning where things go



Use the Map File

Foreshadowing...

Problem	Map Tool
Not enough RAM	Look at summary
Not enough code space	Diff with good map file
Hard fault errors	Find/write viewer
Weird memory errors	Search for address nearby
Planning FW update	Search for variable name
Running too slow	Statistical sampling (hard)

1

Look at Hello.map

TI CCS, CC26XR1

Example hello: prints out “Hello World” to UART

Uses TI’s RTOS

Your .map is probably located where your .hex file is

https://embedded.fm/blog/MapFiles

2

A More Complicated Map File

Hello was 2162 lines long

This one is 14034 lines long

Both TI CCS

3

A Real Memory Map

Ooooh.... I love this part

Use the Map File

Not every tool works for every problem.

Problem	Map Tool
Not enough RAM	Look at summary
Not enough code space	Diff with good map file
Hard fault errors	Find/write viewer
Weird memory errors	Search for address nearby
Planning FW update	Search for variable name
Running too slow	Statistical sampling (hard)
	Read each and every line

Use the Map File: Space

Problem	Map Tool
Not enough RAM	**Look at summary**
Not enough code space	**Diff with good map file**
	Find/write viewer
	Search for address nearby
	Search for variable name
	Statistical sampling (hard)
	Read each and every line

If the map is a wall of impenetrable text, choose a (non-static) global variable or function, one you know is large, and search for it in the map file.

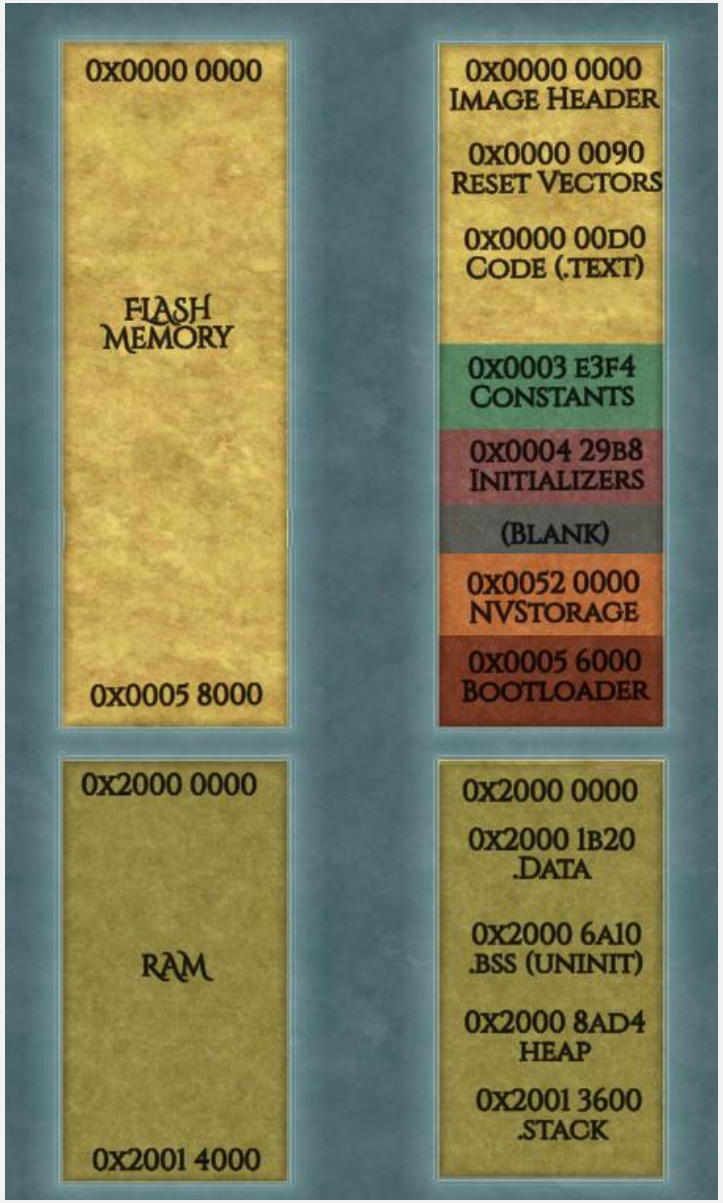
Use the Map File: Space Viewer

Problem

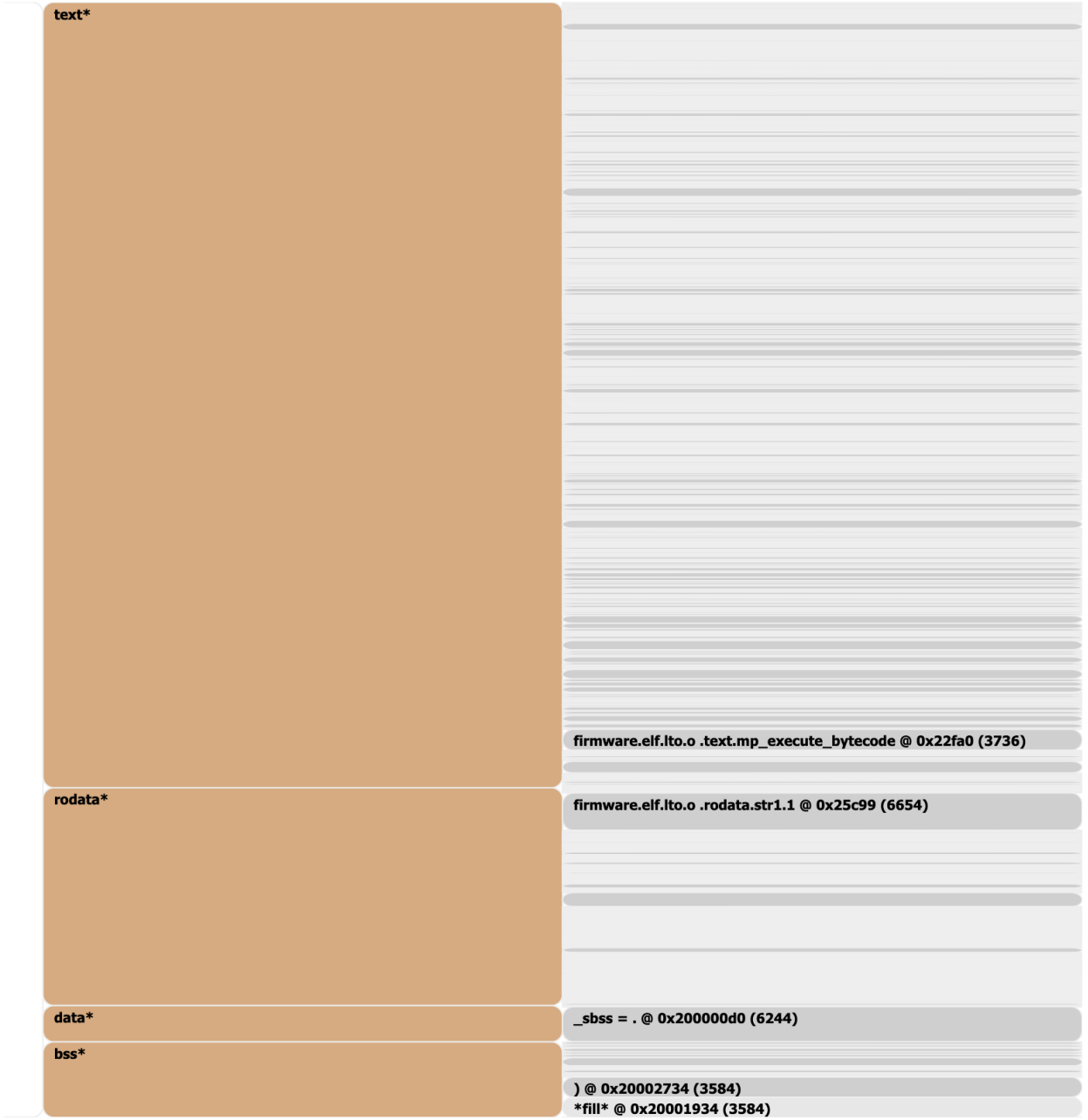
Not enough RAM
Not enough code space

Map Tool

Look at summary
Diff with good map file
Find/write viewer
Search for address nearby
Search for variable name
Statistical sampling (hard)
Read each and every line



Visualizer Example: Circuit Python



Based on github.com/jotux/GccMapVisualizer

Space Optimization Scorecard

Action	Text (code)	Data	Total	Total (hex)	Freed	Total freed
Baseline	31949	324	32273	7E11		
Commented-out test code	26629	324	26953	6949	5320	(Reverted change)
Reimplemented abs()	29845	324	30169	75D9	2104	2104
Calculated const table at init time	29885	244	30129	75B1	40	2144
= comment from you	= size of .text section	= size of .data section	= total image size	= hex of total image size	= bytes freed with this change	= total bytes freed since start

Use the Map File: Debug

Problem	Map Tool
Hard fault errors Weird memory errors	Look at summary **Diff with good map file** Find/write viewer **Search for address nearby** **Search for variable name** Statistical sampling (hard) Read each and every line

Let's talk about debugging the impossible bugs.

You know, those icky, crawly ones that you worry about but can't reliably reproduce.

Use the Map File: Firmware Update

Where, exactly, did I leave the bootloader?

Problem	Map Tool
	Look at summary
	Diff with good map file
	Find/write viewer
	Search for address nearby
Planning FW update	Search for variable name
	Statistical sampling (hard)
	Read each and every line

Firmware Update

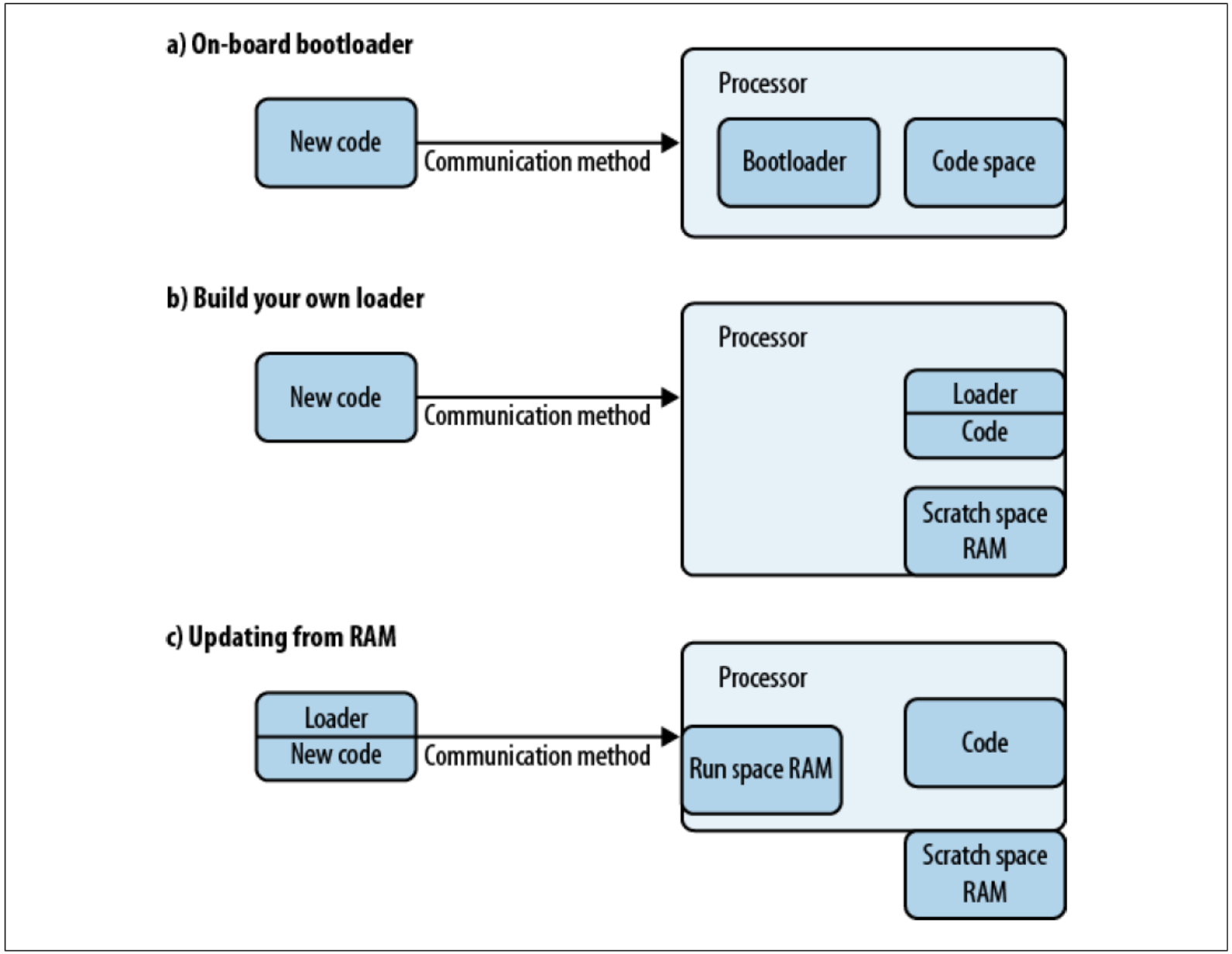


Figure 7-1. Three architectures for loading code

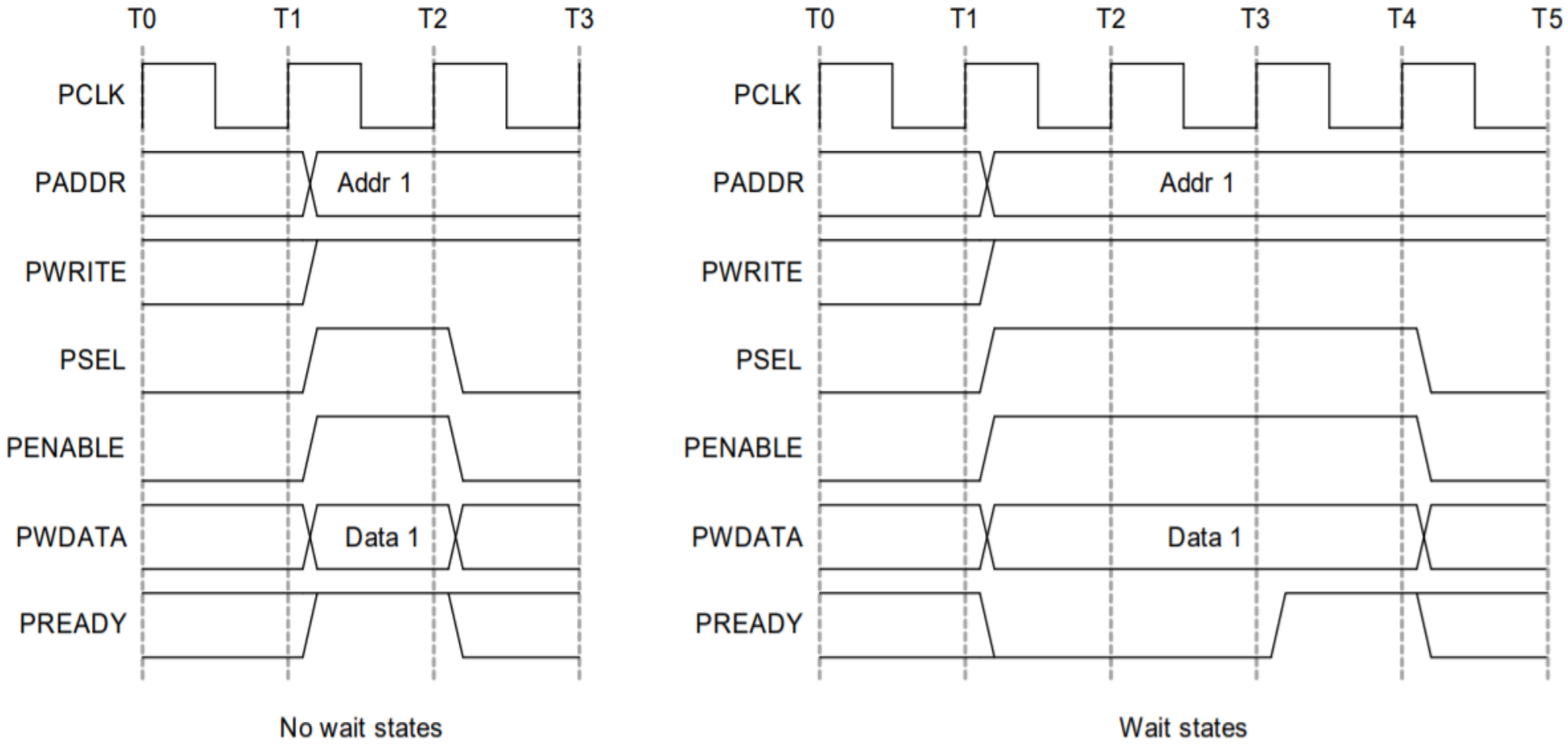
Use a Map File: Too Slow

Problem	Map Tool
	Look at summary
	Diff with good map file
	Find/write viewer
	Search for address nearby
	Search for variable name
Running too slow	**Statistical sampling (hard)**
	Read each and every line

Wait, who is here for the pirate jokes? Why haven't there been any pirate jokes?

Wait State Sadness: Fast CPU and Slow Memory

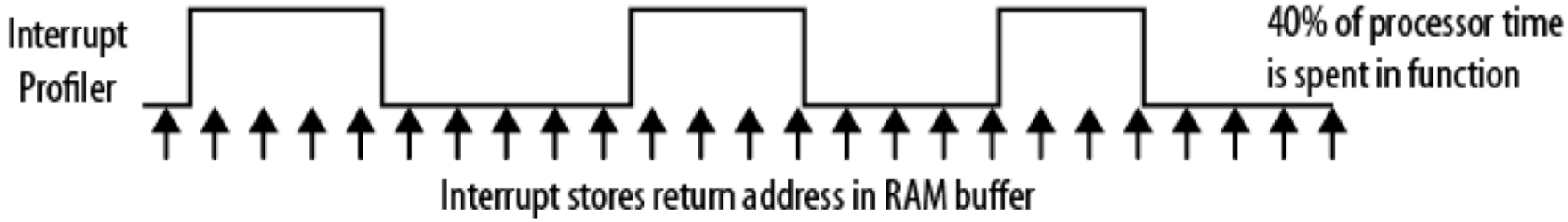
Figure 11-1. APB Write Access



From SAM D21/DA1 Family Datasheet

Statistical Sampling Profiler

What are you doing now? What about now? Now? Now?



Use the Map File

Not every tool works for every problem

Problem	Map Tool
Not enough RAM	Look at summary
Not enough code space	Diff with good map file
Hard fault errors	Find/write viewer
Weird memory errors	Search for address nearby
Planning FW update	Search for variable name
Running too slow	Statistical sampling (hard)
	Read each and every line

Use the Map File

Some solutions are only good as soporifics.

Problem	Map Tool
Not enough RAM	Look at summary
Not enough code space	Diff with good map file
Hard fault errors	Find/write viewer
Weird memory errors	Search for address nearby
Planning FW update	Search for variable name
Running too slow	Statistical sampling (hard)
	~~Read each and every line~~

4

CircuitPython on SAMD21 Map

github.com/adafruit/circuitpython

GCC generated maps are not pretty

Requires linker flags for generation:

LDFLAGS += -Wl,-Map=output.map –Wl,--cref

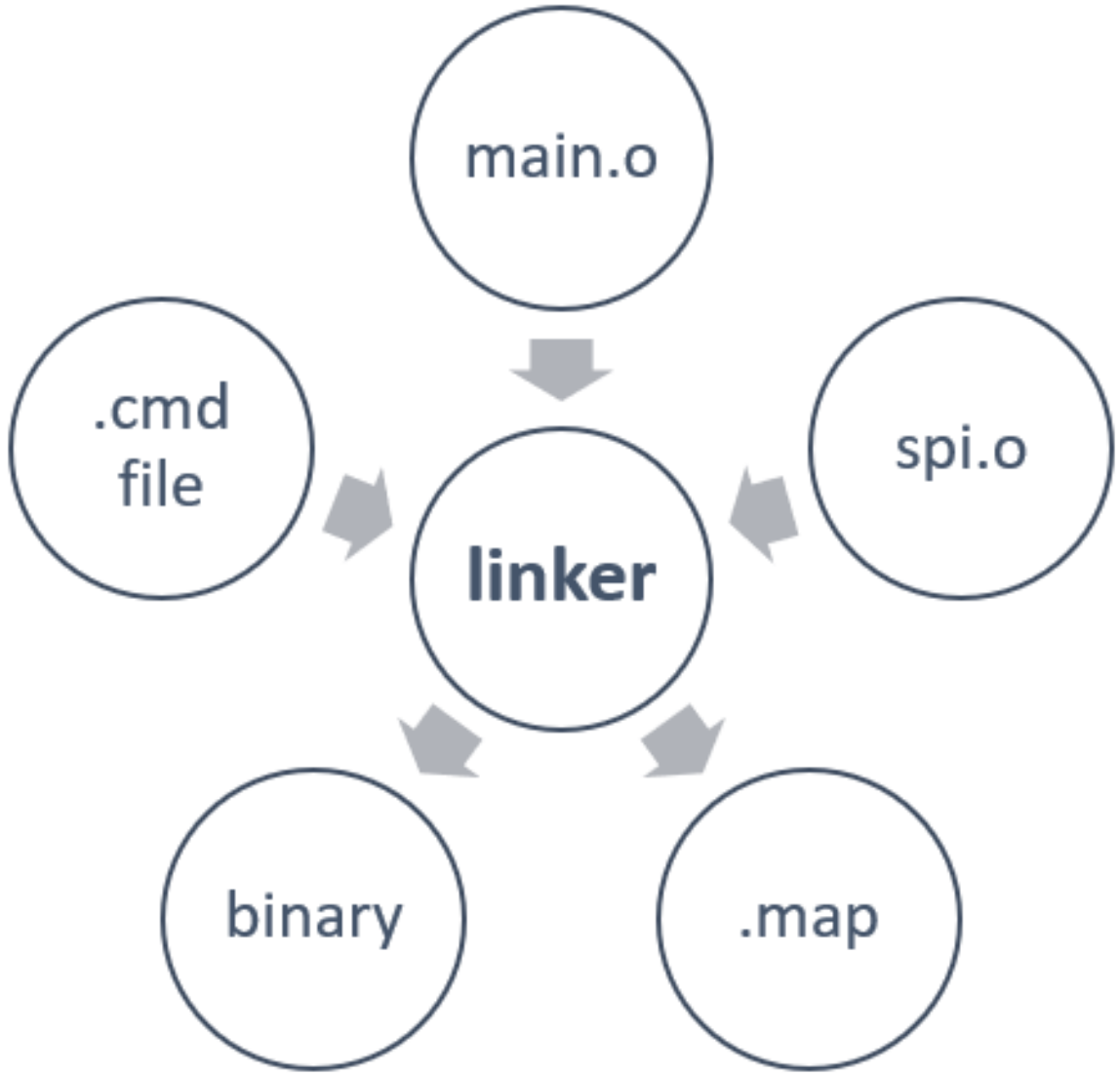
Seeduino XAIO ATSAMD21G18A-MU (ARM Cortex-M0+)

Heap

Everything else is in the heap.

```
5273    .bss.yasmarang_dat
5274                    0x0000000020001930        0x1 firmware.elf.lto.o
5275    *(COMMON)
5276                    0x0000000020001934                . = ALIGN (0x4)
5277    *fill*          0x0000000020001931        0x3
5278                    0x0000000020001934                _ezero = .
5279                    0x0000000020001934                _ebss = .
5280
5281    .stack          0x0000000020001934      0xe00 load address 0x00000000000309b4
5282                    0x0000000020001934                . = ALIGN (0x4)
5283                    0x0000000020002734                . = (. + 0xe00)
5284    *fill*          0x0000000020001934      0xe00
5285                    0x0000000020002734                . = ALIGN (0x4)
5286
```

Where do map files come from?



5

Linker and Map

How did you get to be this way?

ld accepts Linker Command Language files written in a superset of AT&T's Link Editor Command Language syntax.

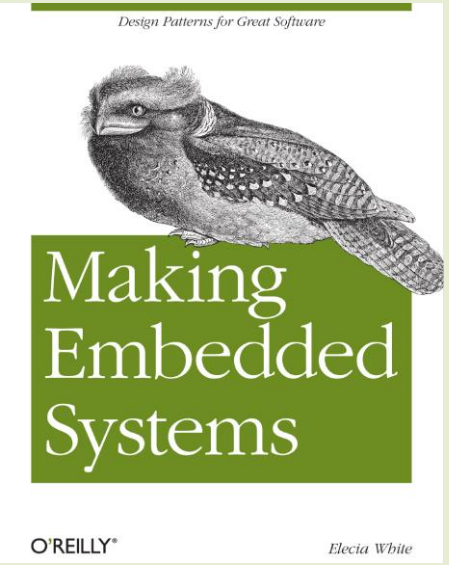
Thank You!

Elecia White

Logical Elegance, Inc.

Embedded https://embedded.fm
https://embedded.fm/blog/MapFiles

Twitter: @logicalelegance



Acknowledgements

All mistakes are my fault, but these people helped make this presentation much better.

Christopher White

Chris Svec

Ben Hencke

Ben Hest

Keith Burzinski

Jacob Beningo

Links

Explore more from these posts:

- Phillip Johnston's Linker-Generated Variables in Libc (Embedded Artistry)
- Thea Flowers' The most thoroughly commented linker script (probably)
- Cyril Fougeray's Get the Most Out of the Linker Map File (at Memfault)
- Govind Mukundan's Analyzing the Linker Map (at EmbeddedRelated)

Memory Map Land created with Inkarnate.com
ARM GCC options https://gcc.gnu.org/onlinedocs/gcc/ARM-Options.html
GNU linker (ld) options man page

Embedded.fm is at https://embedded.fm. It is also available in most podcast apps
Elecia's book is Making Embedded Systems.
She is a co-founder of Logical Elegance, Inc.

Map Visualizers

I’m not endorsing any of these

Puncover: github.com/HBehrens/puncover
Emma: github.com/bmwcarit/Emma
amap: sikorskiy.net/prj/amap/index.html
Bloaty: github.com/google/bloaty
GccMapVisualizer: github.com/jotux/GccMapVisualizer

THANK YOU

Embedded
Online
Conference

www.embeddedonlineconference.com

PRATE!