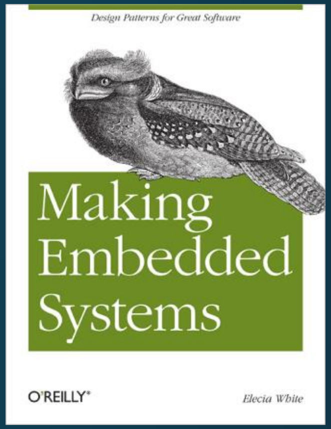
Buried Treasure and Map Files

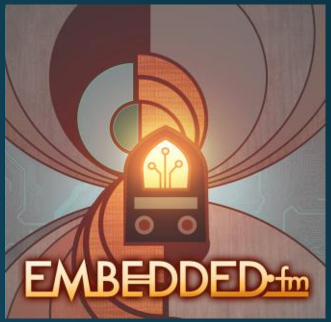
Ahoy there, matey!


Design Patterns for Great Software

Making Embedded Systems

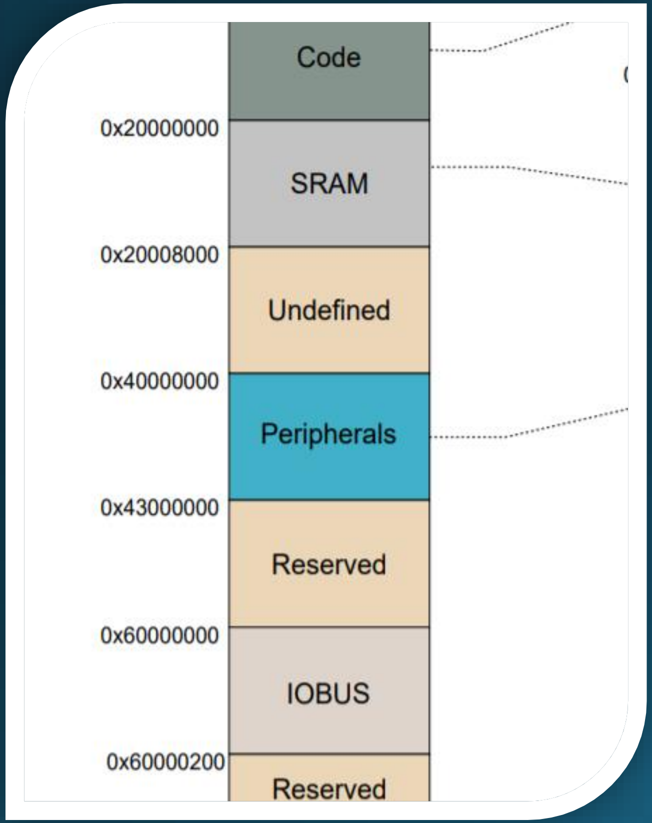
O'REILLY

Elecia White

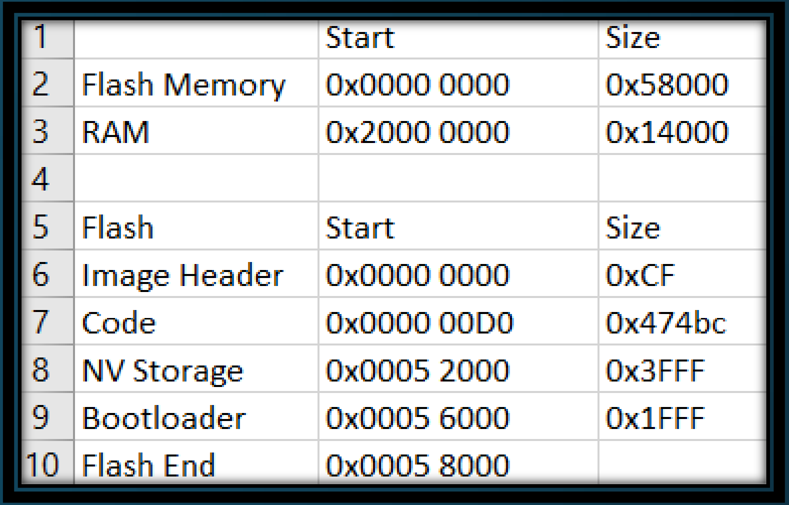


Memory Maps

There are many ways of looking at memory



STATIC SHORES
GLOBAL GROVES
.DATA
.RAMFUNCS
.RAMFUNCS
INITOPIA
FLASH FEDERATION
NULL WHIRLPOOL
COUNTY OF CODE
LAKE OF ZEROING
.BSS
RIVER OF INITIALIZATION
.CINIT
CONSTANTINIA
RAM LANDIA
MEMORY CONFIGURATION MOUNTAINS
.CONST
.RODATA
MIRROR LAKE
FOREST OF FUNCTIONS
ORCHARD OF FUNCTIONS
STACKS
.SWITCH
SUMMARY VALLEY
HEAP HALLOWS
NONVOLATILE STORAGE
.TEXT
PORT OF PERIPHERAL REGISTERS
OCEAN OF UNUSED ADDRESS SPACE
BOOTLOADER
.VECTOR
FJORDS OF DISCARDED FUNCTIONS
MEMORY MAP LAND
CREATED BY ELECIA WHITE, LOGICAL ELEGANCE, INC.
SOURCE AT EMBEDDED.FM/BLOG/MAPFILES
UNCHARTED REGISTERS FILLED WITH MYSTERIOUS VALUES
KINGDOM OF SPI FLASH

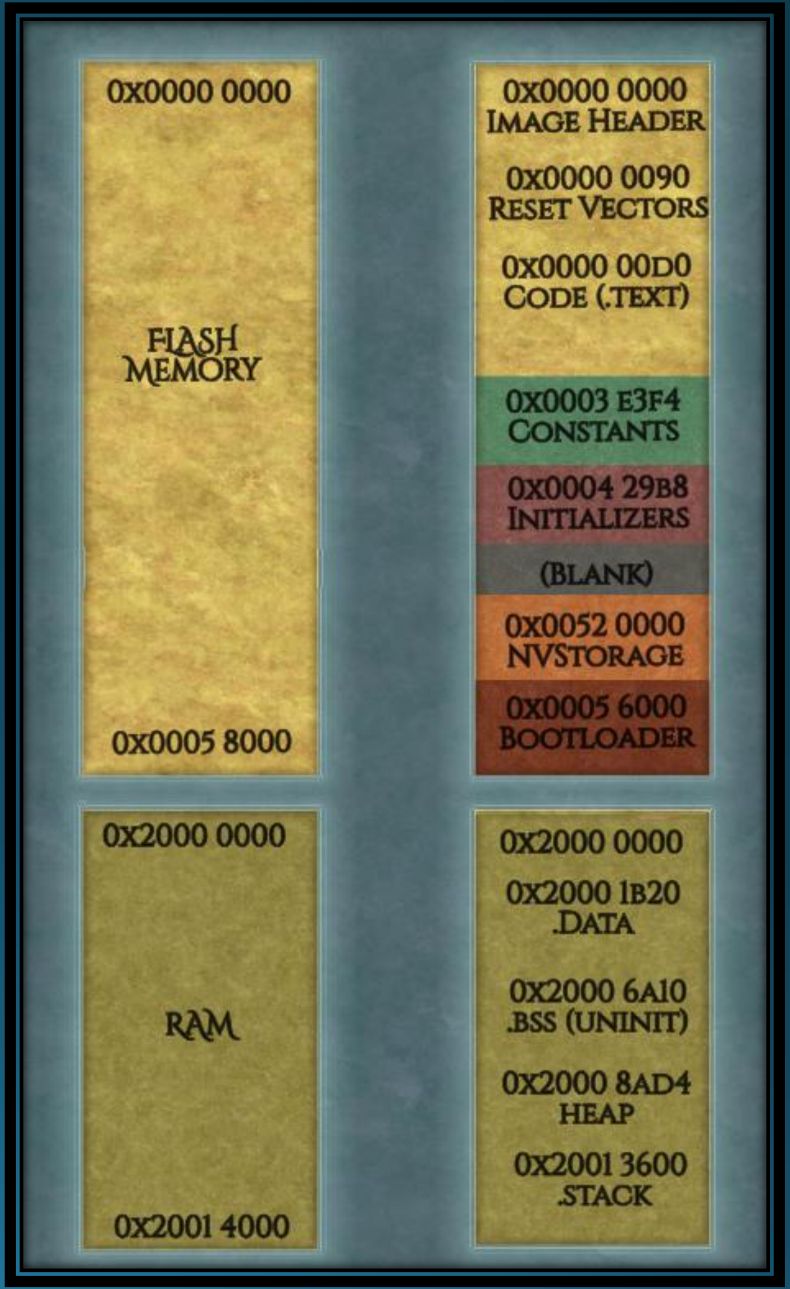


1		Start	Size
2	Flash Memory	0x0000 0000	0x58000
3	RAM	0x2000 0000	0x14000
4			
5	Flash	Start	Size
6	Image Header	0x0000 0000	0xCF
7	Code	0x0000 00D0	0x474bc
8	NV Storage	0x0005 2000	0x3FFF
9	Bootloader	0x0005 6000	0x1FFF
10	Flash End	0x0005 8000	

Elecia White, Logical Elegance, Inc. https://embedded.fm/blog/MapFiles

Planning where things go

Memory Layout



0x0000 0000

FLASH MEMORY

0x0005 8000

0x0000 0000
IMAGE HEADER

0x0000 0090
RESET VECTORS

0x0000 00D0
CODE (.TEXT)

0x0003 E3F4
CONSTANTS

0x0004 29B8
INITIALIZERS

(BLANK)

0x0052 0000
NVSTORAGE

0x0005 6000
BOOTLOADER

0x2000 0000

RAM

0x2001 4000

0x2000 0000

0x2000 1B20
.DATA

0x2000 6A10
.BSS (UNINIT)

0x2000 8AD4
HEAP

0x2001 3600
.STACK

Elecia White, Logical Elegance, Inc. https://embedded.fm/blog/MapFiles

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)
	Read each and every line

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

A More Complicated Map File

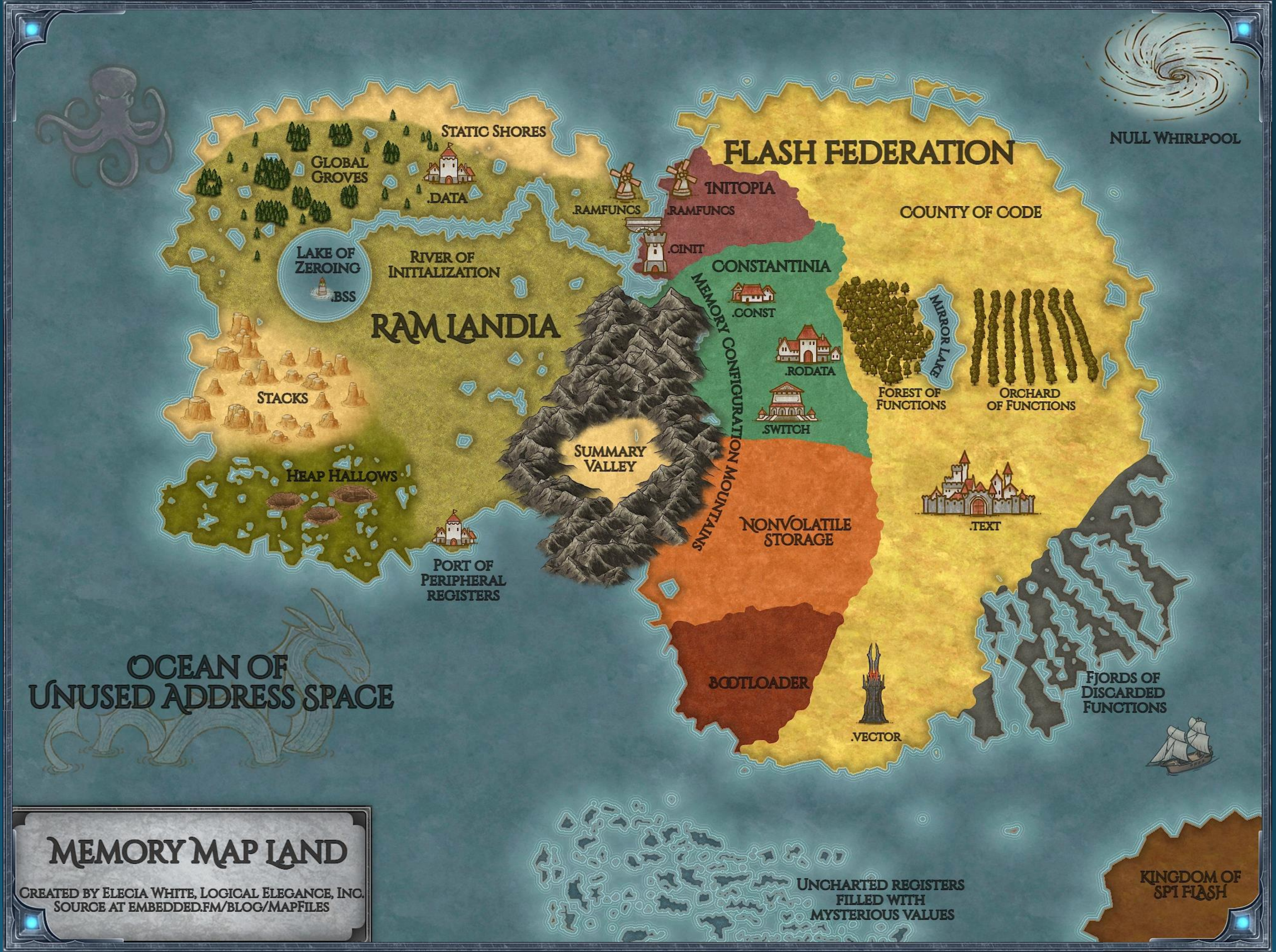
Hello was 2162 lines long

This one is 14034 lines long

Both TI CCS

A Real Memory Map

Ooooh…. I love this part



Elecia White, Logical Elegance, Inc. https://embedded.fm/blog/MapFiles

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

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

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

Problem

Not enough RAM

Not enough code space

Map Tool

Look at summary

Diff with good map file

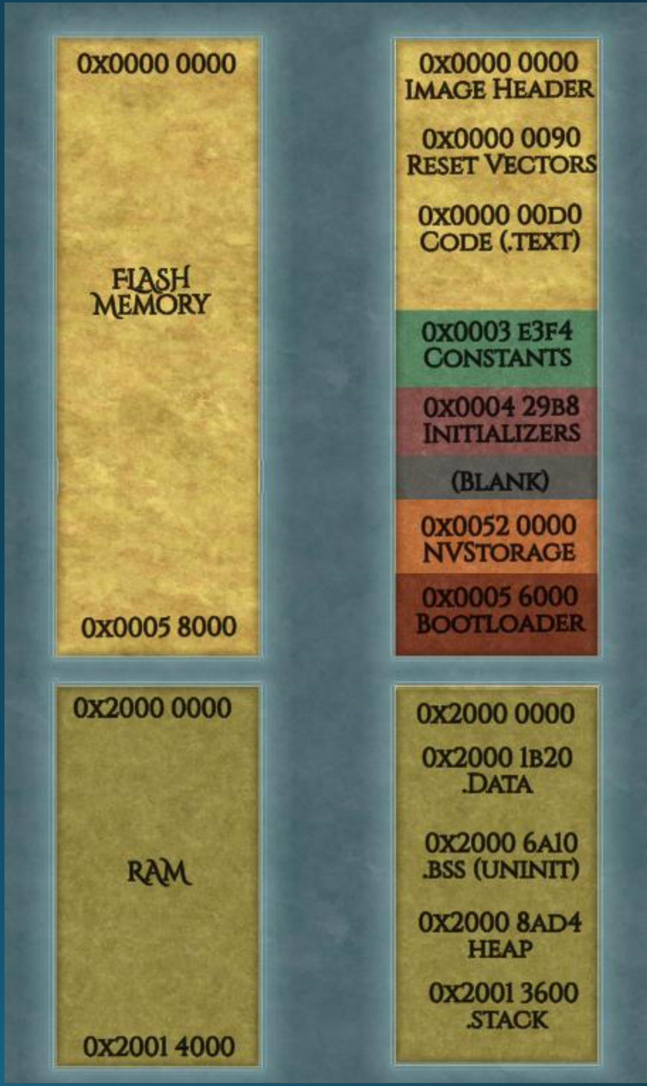
Find/write viewer

Search for address nearby

Search for variable name

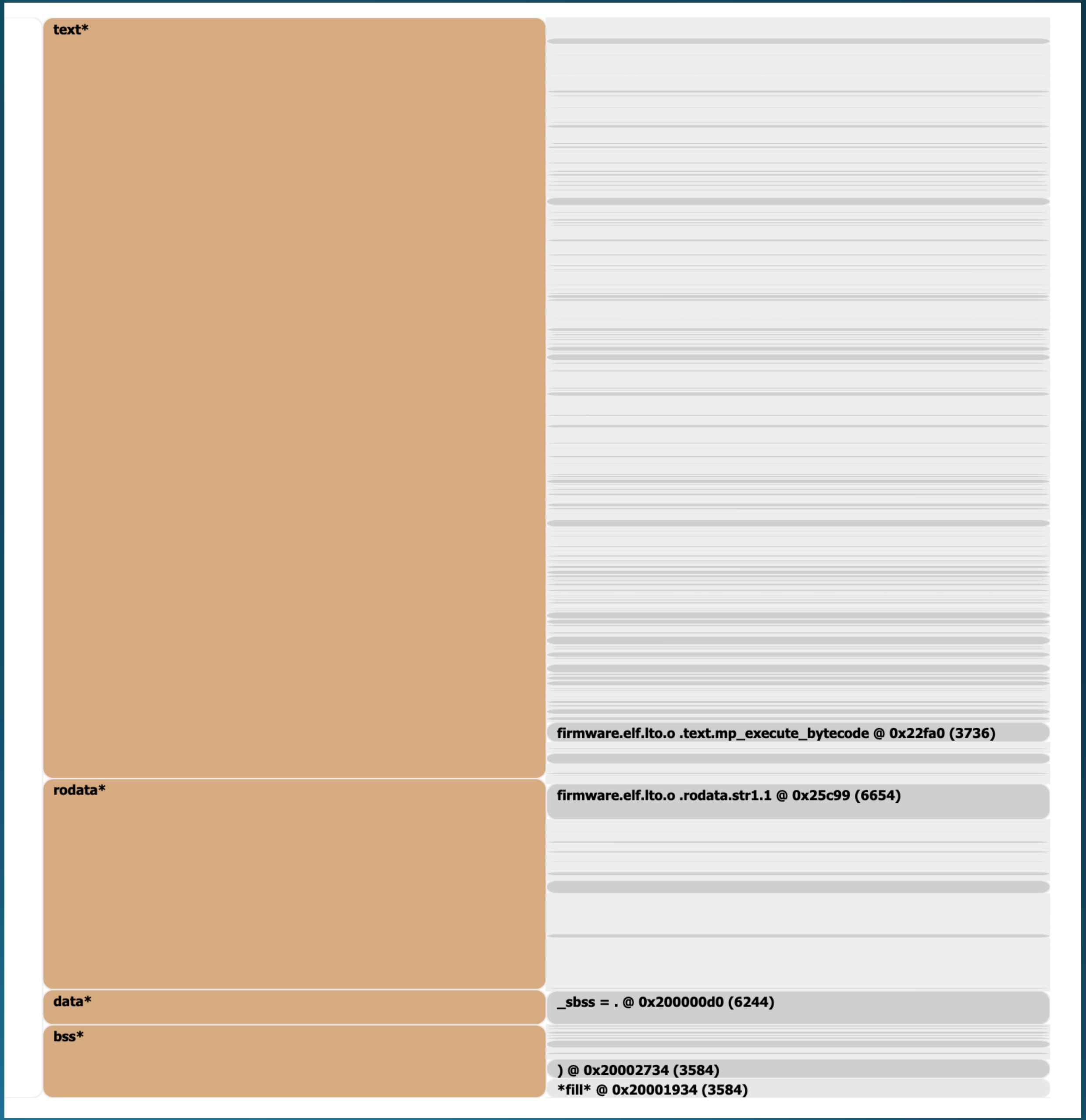
Statistical sampling (hard)

Read each and every line

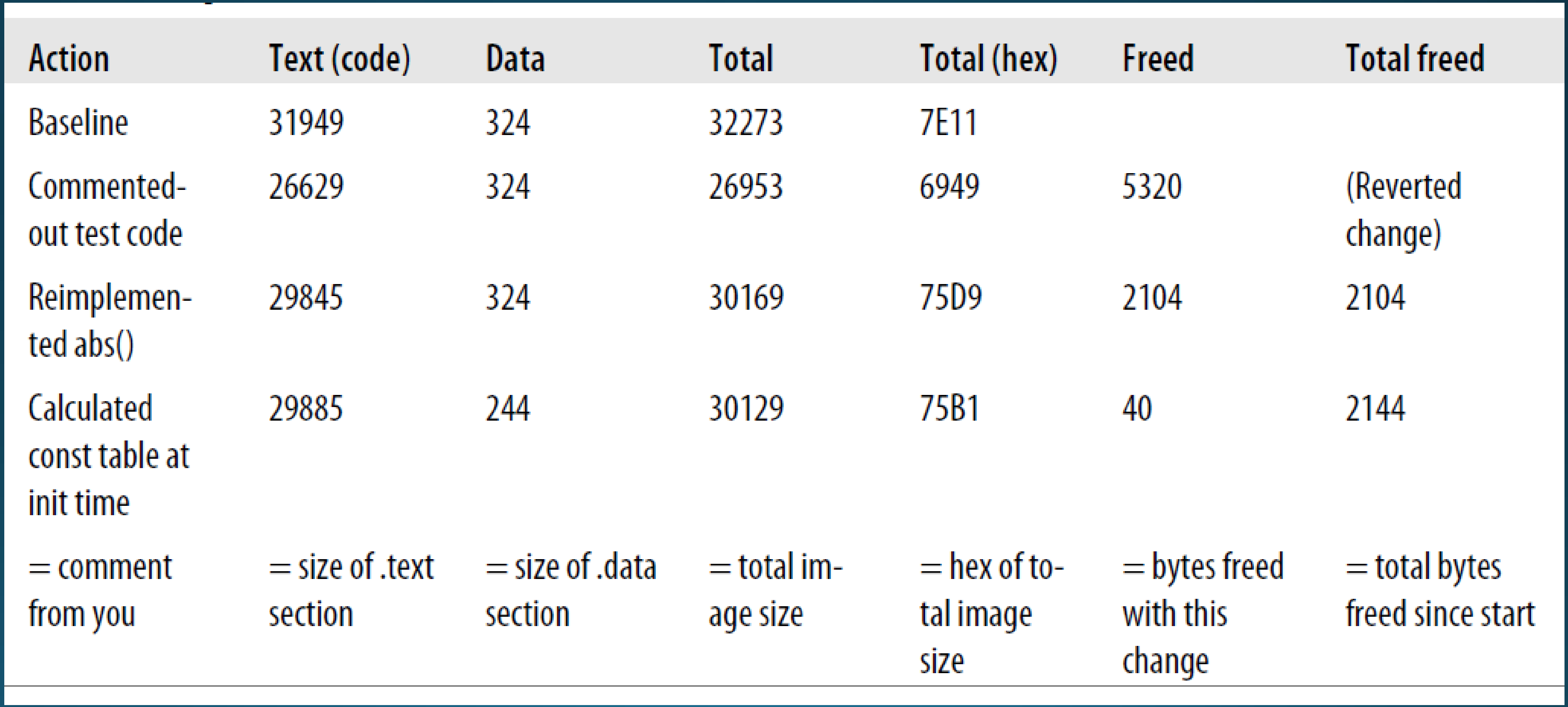


Elecia White, Logical Elegance, Inc. https://embedded.fm/blog/MapFiles

Visualizer Example: Circuit Python


text*
firmware.elf.lto.o .text.mp_execute_bytecode @ 0x22fa0 (3736)
rodata*
firmware.elf.lto.o .rodata.str1.1 @ 0x25c99 (6654)
data*
_sbss = . @ 0x200000d0 (6244)
bss*
) @ 0x20002734 (3584)
fill @ 0x20001934 (3584)

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

Elecia White, Logical Elegance, Inc. https://embedded.fm/blog/MapFiles

Use the Map File

Problem	**Map Tool**
	Look at summary
	Diff with good map file
Hard fault errors	Find/write viewer
Weird memory errors	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

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

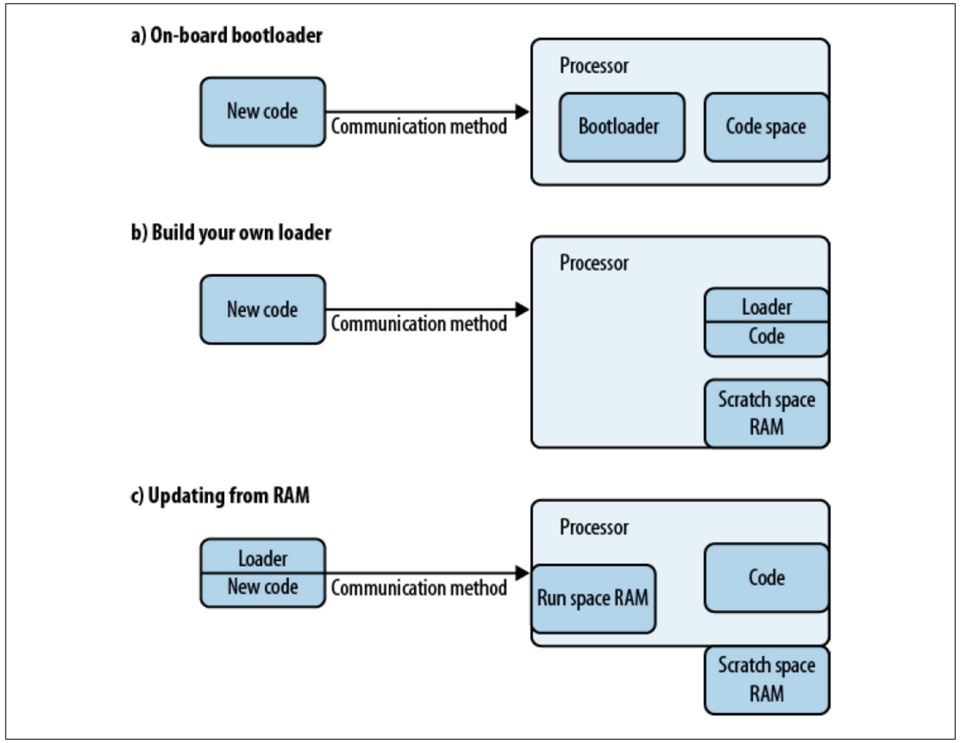


Figure 7-1. Three architectures for loading code

Elecia White, Logical Elegance, Inc. https://embedded.fm/blog/MapFiles

Use the Map File

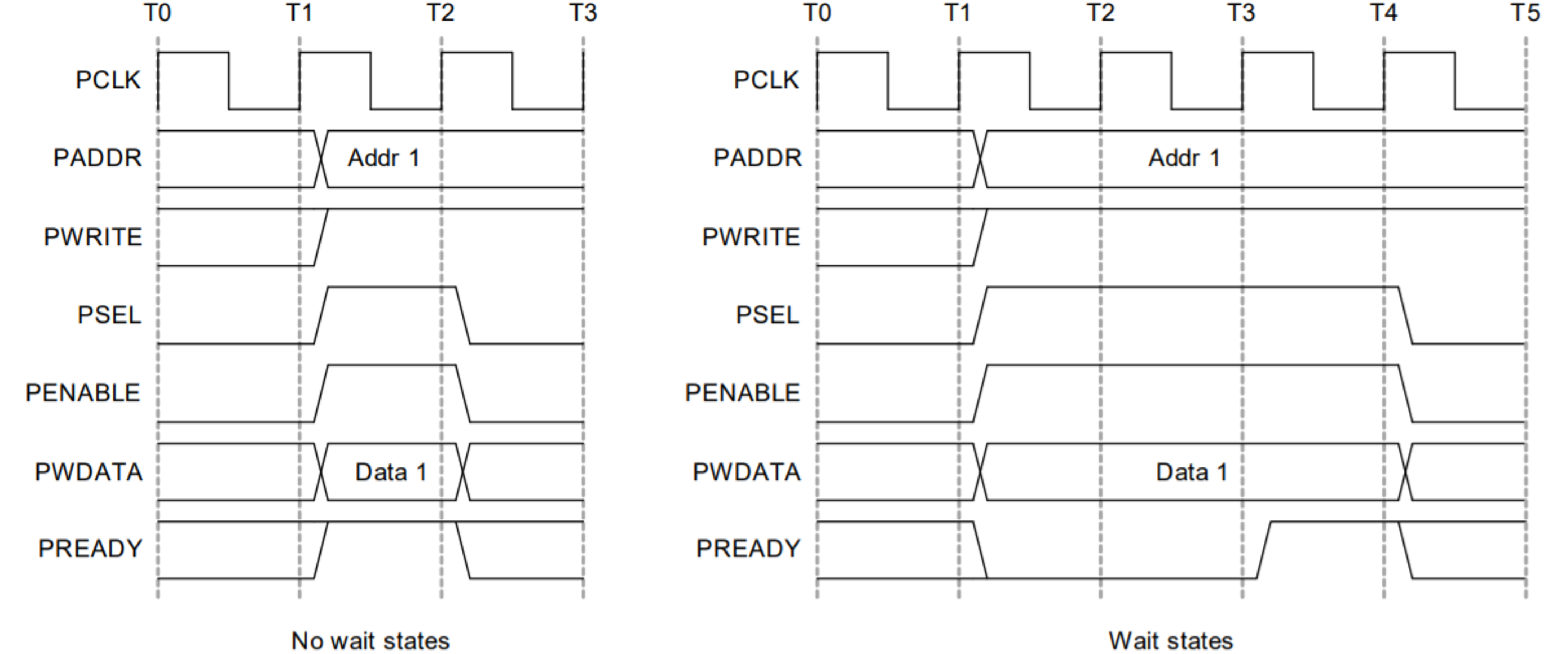
Wait, who is here for the pirate jokes? Why haven't there been any pirate jokes?

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 State Sadness

Fast CPU and Slow Memory

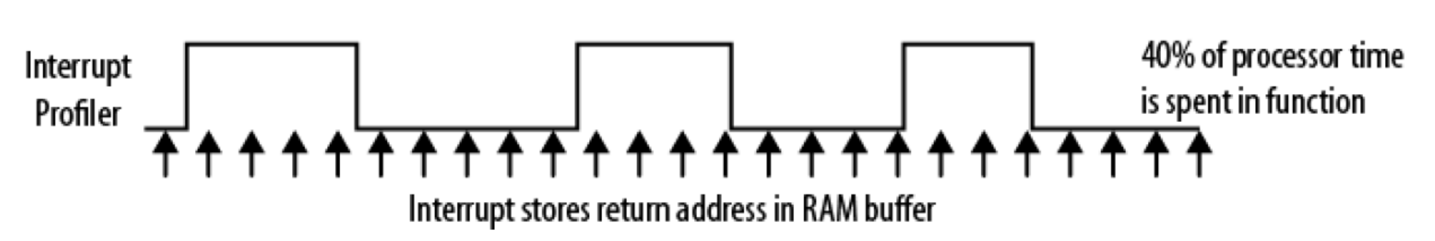
Figure 11-1. APB Write Access



Elecia White, Logical Elegance, Inc. https://embedded.fm/blog/MapFiles

Statistical Sampling Profiler

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


Interrupt Profiler
40% of processor time is spent in function
Interrupt stores return address in RAM buffer

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~~

CircuitPython on SAMD21 Map

github.com/adafruit/circuitpython

GCC generated maps are not pretty

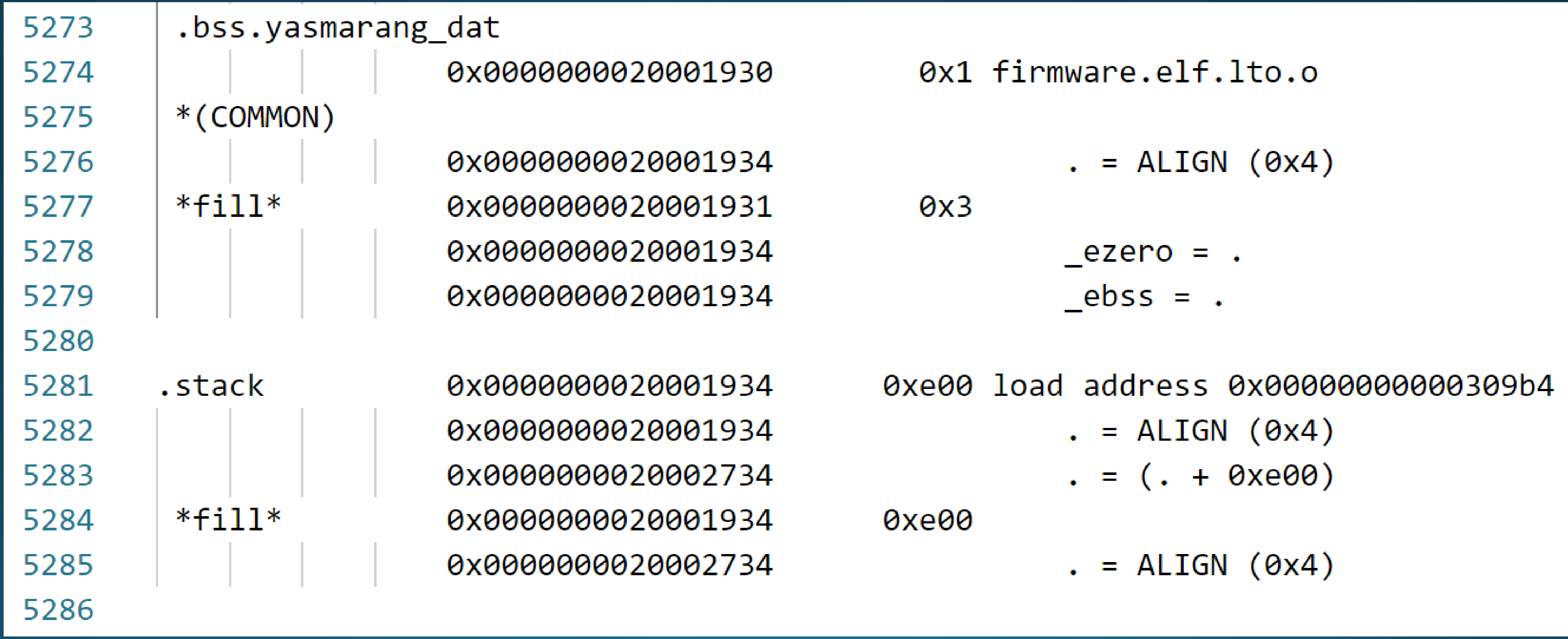
Requires linker flags for generation:

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

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

Heap

Everything else is in the heap

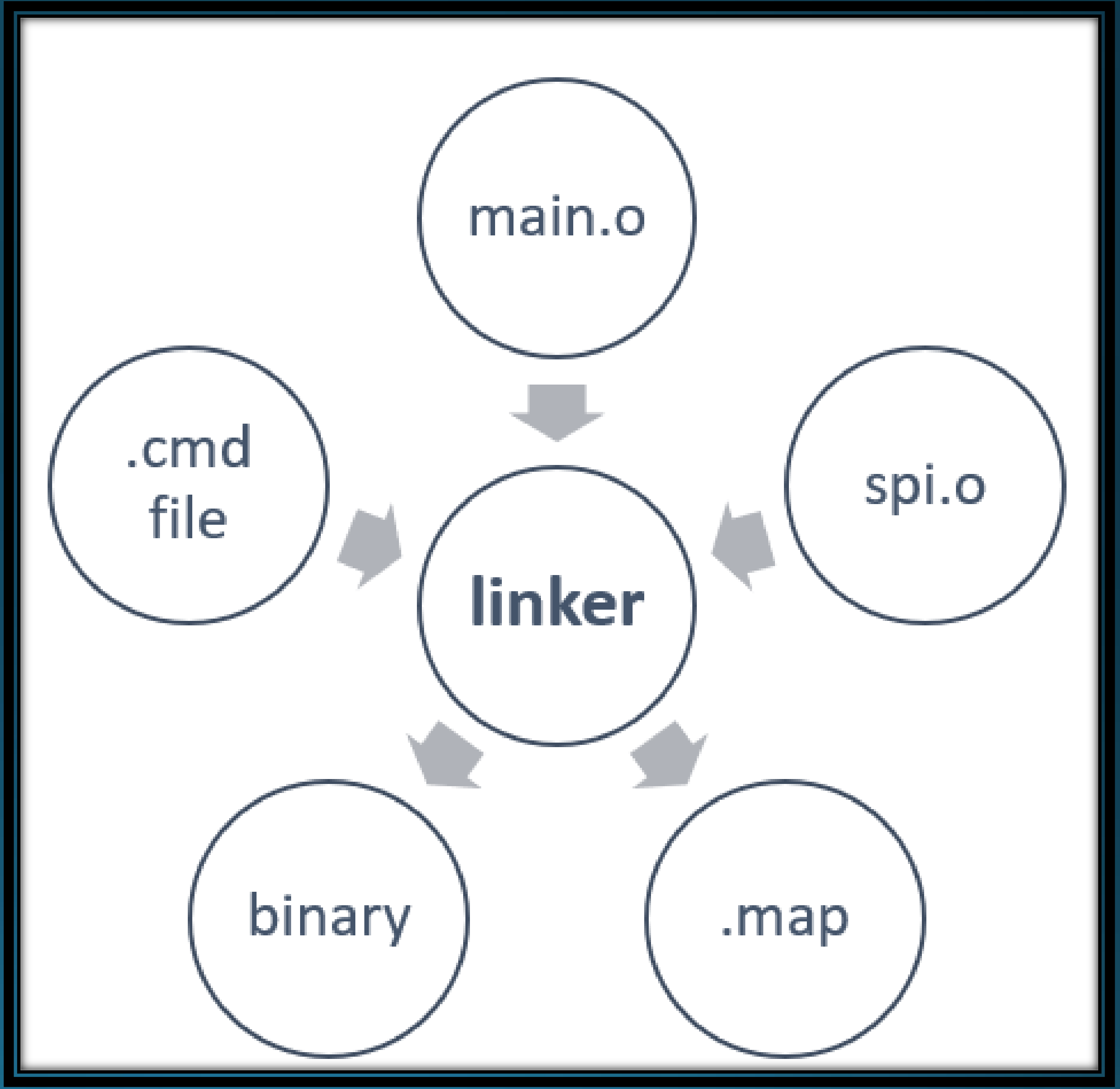


```
  .bss.yasmarang_dat
                0x0000000020001930        0x1 firmware.elf.lto.o
  *(COMMON)
                0x0000000020001934                . = ALIGN (0x4)
  *fill*        0x0000000020001931        0x3
                0x0000000020001934                _ezero = .
                0x0000000020001934                _ebss = .

.stack          0x0000000020001934      0xe00 load address 0x00000000000309b4
                0x0000000020001934                . = ALIGN (0x4)
                0x0000000020002734                . = (. + 0xe00)
  *fill*        0x0000000020001934      0xe00
                0x0000000020002734                . = ALIGN (0x4)
```

Elecia White, Logical Elegance, Inc. https://embedded.fm/blog/MapFiles

Where do map files come from?



Elecia White, Logical Elegance, Inc. https://embedded.fm/blog/MapFiles

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

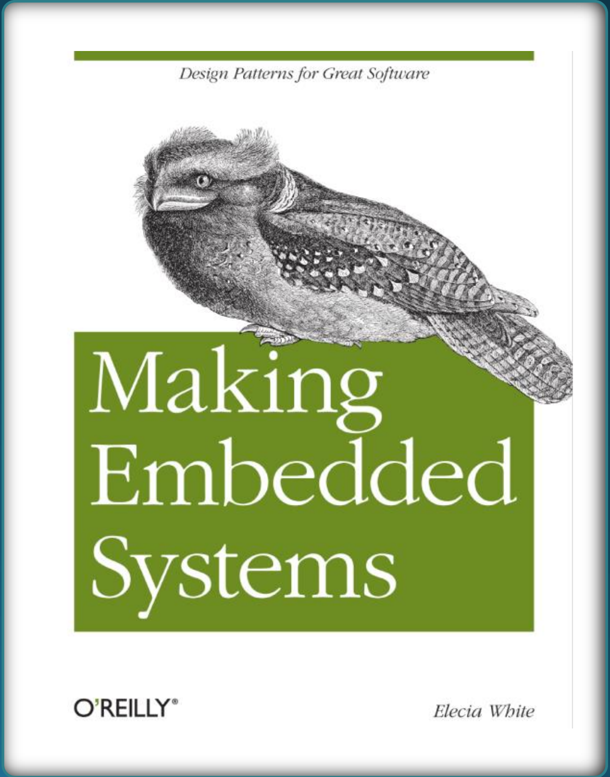
Logical Elegance, Inc.

Embedded https://embedded.fm

https://embedded.fm/blog/MapFiles

Twitter: @logicalelegance

EMBEDDED.fm



Design Patterns for Great Software

Making Embedded Systems

O'REILLY®

Elecia White

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!

Goodbye…

Talk originally prepared for
Embedded Online Conference 2021

https://www.embeddedonlineconference.com/

PRATE!