

1. What is the Endianness ?
2. Explain how interrupts work. What are some things that you should never do in an interrupt function?
3. Explain the interrupt vector table.
4. Explain when you should use "volatile" in C.
5. Explain UART, SPI, I2C buses. Describe some of the signals in each. At a high-level describe each.
6. Explain how DMA works. What are some of the issues that you need to worry about when using DMA?
7. Where does the interrupt table reside in the memory map for various processor families?
8. In which direction does the stack grow in various processor families?
9. Implement a Count Leading Zero (CLZ) bit algorithm, but don't use the assembler instruction.

What optimizations to make it faster? What are some uses of CLZ?

10. What is RISC-V? What are its claimed pros or cons?
11. List some ARM cores. For embedded use, which cores have which architecture (von-neumann or Harvard).
12. Explain processor pipelines, and the pro/cons of shorter or longer pipelines.
13. Explain memory segments (ram memory).
14. What is a pull-up or pull-down resistor? When might you need to use them?
15. What is "zero copy" or "zero buffer" concept?
16. How do you determine if a memory address is aligned on a 4 byte boundary in C?
17. What hardware debugging protocols are used to communicate with ARM microcontrollers?
18. What is padding and packing in memory (used usually with structs in C)?

19. What are the basic concepts of what happens before `main()` is called in C - (start up process in microprocessor/microcontroller) ?
20. What are the basic concepts of how `printf()` works? List and describe some of the special format characters? Show some simple C coding examples.
21. Describe each of the following? SRAM, Pseudo-SRAM, DRAM, ROM, PROM, EPROM, EEPROM, MRAM, FRAM, flash memory.
22. Show how to declare a pointer to constant data in C. Show how to declare a function pointer in C.
23. How do you multiply without using multiply or divide instructions for a multiplier constant of 10, 31, 132?
24. When do you use `memmove()` instead of `memcpy()` in C? Describe why.
25. Why is `strlen()` sometimes not considered "safe" in C? How to make it safer? What is the newer safer function name?

26. When is the best time to malloc() large blocks of memory in embedded processors? Describe alternate approach if malloc() isn't available or desired to not use it, and describe some things you will need to do to ensure it safely works.
27. Describe symbols on a schematic? What is a printed circuit board?
28. Do you know how to use a logic probe? multimeter? oscilloscope? logic analyzer? function generator? spectrum analyzer? other test equipment? Describe when you might want to use each of these?
29. What processors or microcontrollers are considered 4-bit? 8-bit? 16-bit? 24-bit? 32-bit?
30. What is ohm's law?
31. What is Nyquist frequency (rate)? When is this important?
32. What is "wait state" of a task or process?
33. What is the difference between process and task ?

34. What are some common logic families?
35. What is a CPLD? an FPGA? Describe why they might be used in an embedded system?
36. List some types of connectors found on test equipment.
37. What is the daemon process? why do we need it?
38. What is RS232? RS432? RS485? MIDI? What do these have in common?
39. What is ESD? Describe the purpose of "pink" ESD bags? black or silvery ESD bag? How do you properly use a ground strap? When should you use a ground strap? How critical is it to use ESD protections? How do you safely move ESD-sensitive boards between different parts of a building?
40. What is "Lockout-Tagout"?
41. What is ISO9001? What is a simple summary of it's concepts?

42. What is A/D? D/A? OpAmp? Comparator Other Components Here? Describe each. What/when might each be used?
43. What is the difference between debugger and in-circuit emulator?
44. What embedded RTOS have you used? what are the properties of a good RTOS?
45. Why printf() is not friendly to be called in ISR (interrupt service routine) ?
46. Have you ever used any encryption algorithms? Did you write your own from scratch or use a library (which one)? Describe which type of algorithms you used and in what situations you used them?
47. What is a CRC algorithm? Why would you use it? What are some CRC algorithms? What issues do you need to worry about when using CRC algorithms that might cause problems?
48. List the interprocess communication types?

49. What segment of memory is used to store string values?
50. What issues are a concern for algorithms that read/write data to DRAM instead of SRAM?
51. what is the use of static keyword in C?
52. What are the compilation process stages in C?
53. After powerup, have you ever initialized a character display using C code? From scratch or library calls?
54. Have you ever written a RAM test from scratch?
What are some issues you need to test?
55. Have you ever written code to initialize (configure) low-power self-refreshing DRAM memory after power up (independent of BIOS or other code that did it for the system)? It's likely that most people have never done this.
56. Write code in C to "round up" any number to the next "power of 2", unless the number is already a power of 2. For example, 5 rounds up to 8, 42

rounds up to 64, 128 rounds to 128. When is this algorithm useful?

- 57. What are two of the hardware protocols used to communicate with SD cards? Which will most likely work with more microcontrollers?
- 58. What issues concerns software when you WRITE a value to EEPROM memory? FLASH memory?
- 59. What is NOR-Flash and NAND-Flash memory? Are there any unique software concerns for either?
- 60. What is the PLL?
- 61. What's the use of Fork() function in Linux?
- 62. What is the difference between semaphore and mutex ?
- 63. What is the booting stages of embedded linux OS?
- 64. What is the interrupt vector table?
- 65. What do we need initial program loader before the bootloader (u-boot for example) ?

66. What is the racing condition in shared resources?
67. What is the device tree in Linux?
68. How does Scatter/Gather DMA differ from "normal" DMA?
69. What does connect() do on a UDP socket?
70. What is the difference between .elf, .hex and .bin executables?
71. What is the extern keyword in C?
72. Describe ethernet protocols (TCP/IP) and differ it from (UDP).
73. What is the most common protocol is used to communicate with a "smart card"? Have you ever written any software to communicate with a "smart card" in an embedded product?
74. What is I2S? Where is it used? Why might you want to use I2S in an embedded system?
75. What is CAN, LIN, FlexRay? Where are they used? Have you ever used any?

76. What is the difference between Router and switch?
77. What in-circuit debuggers or programmers have you used?
78. What does the assembler do?
79. What does the linker do? does it allocate memory in this stage?
80. What is dual-port RAM? Why would it be useful in some embedded systems?
81. Mention HW and SW debugging type for embedded linux os?
82. If you create a circular buffer, what size of buffer might optimized code be slightly faster to execute? why?
83. Atomic instructions, give examples ?
84. What is a DSP?
85. What are virtual and physical addresses? What is a MMU?

86. Describe different types of Cache. When do you need to flush the cache, when to invalidate cache?
87. What is SIMD?
88. What is a Mailbox register?
89. What is a Cacheline?
90. What is a Mutex?
91. What is Scatter-Gather DMA? What is Ping-Pong DMA?
92. What is Prescaler in timers?
93. How to determine the accuracy of a ADC?
94. Static vs dynamic allocation: pros, cons?
95. What is NULL pointer and what is its use?
96. What is void pointer and what is its use?
97. What is interrupt latency?

98. Advantages and disadvantages of using macro and inline functions? What happens when recursive functions are declared inline?
99. Is Count Down_to_Zero Loop better than Count_Up_Loops?
100. What is priority inversion? How to solve it?