

```

This file has values separated by tabs.
"When opening with a spreadsheet program, specify that tabs are delimiters."
"When saving (e.g. with Excel), save as ""Text (Tab delimited) (*.txt)"" "
# Response (may use cells in adjacent columns if needed)
1 wuf4@ncsu.edu
2 22108 or 22084
3 11144
4 6176
5 348
6 17072 or 17056
7 1260 or 1262
8 _printf_core : 768 bytes
9 os_mem : 4096
10 Control_HBLED : 560 or 566
11 Lucida_Console12x19 : 3831
12 st7789.0 : 1900
13 lucida_12x19.o : 3831
14 rtx_lib.o : 4940
15 Control.o : 3968
16 rtx_thread.o : 2038 or 2044
17 rtx_kernel.o : 11
18 rtx_kernel.o : 164
19 os_systick : 1
20 _printf_core : 768, Control_HBLED : 560, svcRTxThreadnew : 484 or 492
21 yes
22 no
23 [Unclear if there is a way to do this, so look for any solutions offered].
24 Lucida_Console12x19 : 3831, g_meas_sample : 1920, g_set_sample : 1920
25 yes
26 0x00002500 4904 LDR r1,[pc,#16] or 0x000024F4 4904 LDR r1,[pc,#16]
27 No, it needs to be on a 2 byte boundary
28 0x00002512 4770 BX lr or 0x00002506 4770 BX lr
29 It is defining the lower half-word of the constant 10000 (0x00002710)
30 It is defining the upper half-word of the constant 10000 (0x00002710)
31 multiplies the t variable by 10000
32 r0
33 it is a word at address 0x00002514 (or 0x00002508). This value is loaded by
the ldr at 0x00002500 (or 0x000024f4) into register r1. MULS multiplies r1 and r0.
34 Stores the t variable to memory. The volatile keyword makes it write to
memory always.
35 0x00002500,0x00002504,no 0x00002506,0x0000250A,yes-branch
0x0000250C,0x0000250E,yes-branch 0x00002510,0x00002512,yes-return
35 0x000024f4,0x000024f8,no 0x000024fa,0x000024fe,yes-branch
0x00002500,0x00002502,yes-branch 0x00002504,0x00002506,yes-return
38 No. There is a lot of conditional control flow in the CFG, but not in the
source code.
39 No
40 No
41 Test_Fault get inlined into Thread_Fault_Injector, and the original
Test_Fault function code is removed by the linker.

```

```

42     ADD pc,pc,r0
43     r0, Arm parameter passing convention
45     Each entry is one byte. There are 14 entries, arg contains the test number,
ie the location in the table. LDRB loads the jump offset (in half-words) from the
table. The jump offset is doubled (to get bytes) and is then added to the pc
register. This will cause the pc the jump to the case's code.
46     112 bytes, the table is 14 bytes, so there is more then enough space.
50     In some TR_Fill_Queue, there is no exit, so there is an infinte loop. So
there would be no need to have a jmp to the merge point in the switch statement.
51     no and yes, two of the case are the same. so the compiler would have merged
these to into one.
52     The linker listing says that no function calls main
53.a   osKernelStart, osKernelInitialize, Sound_Disable_Amp, LCD_Text_PrintStr_RC,
LCD_Text_Init, LCD_Init, LCD_Erase, Fault_Init, Create_OS_Objects, Init_RGB_LEDs,
Control_RGB_LEDs, Delay, Init_Debug_Signals, Init_Buck_HBLED
53.b   UI_Process_Touch, LCD_TS_Read, osDelay
55.a   108
55.b   44
55.c   132
55.d   120 or 104
55.e   200
55.f   0
56.a   ADC0_IRQHandler, 176
56.b   DMA0_IRQHandler, 64
56.c   PIT_IRQHandler, 8
56.d   PendSV_Handler, 56
56.e   SysTick_Handler , 80
56.f   TPM0_IRQHandler, 4
57     __aeabi_fsub, __aeabi_frsub, Control_HBLED
58     Control_HBLED, ADC0_IRQHandler
59     _float_epilogue, _float_round
61
62     yes
64     The compiler turns the recursion into a loop. So loop iterations do not use
additional stack space.
65     No. Instead it is adding two. It is not clear why. However, the return value
may not matter, since the function never returns.
67     Example differences: No variable declarations. Some symbols (e.g. variable
names, macros) are missing in decompiled version. Decompiled version uses variable
names which describe memory locations instead. If/else case layout flipped. No
preprocessor directives. Conditionally compiled code which wasn't compiled isn't
included (#if USE_SYNC_NO_FREQ_DIV).

```