# Static Call Graph for image .\Objects\NewRTOSProject.axf

#<CALLGRAPH># ARM Linker, 5060750: Last Updated: Mon Oct 22 16:59:31 2018

### Maximum Stack Usage = 120 bytes + Unknown(Functions without stacksize, Cycles, Untraceable Function Pointers)

### Call chain for Maximum Stack Depth:

svcRtxKernelStart ⇒ osRtxThreadStartup ⇒ svcRtxThreadNew ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

### Functions with no stack information

* [\_\_user\_initial\_stackheap](#2nusc19)

### Mutually Recursive functions

* [NMI\_Handler](#46r0co2)   ⇒   [NMI\_Handler](#46r0co2)

### [HardFault\_Handler](#2lwamvv)   ⇒   [HardFault\_Handler](#2lwamvv) Function Pointers

* + [ADC0\_IRQHandler](#111kx3o) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [CMP0\_IRQHandler](#3l18frh) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [DAC0\_IRQHandler](#206ipza) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [DMA0\_IRQHandler](#4k668n3) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [DMA1\_IRQHandler](#2zbgiuw) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [DMA2\_IRQHandler](#1egqt2p) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [DMA3\_IRQHandler](#3ygebqi) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [DefaultISR](#2dlolyb) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(.text)
  + [FTFA\_IRQHandler](#sqyw64) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [HardFault\_Handler](#2lwamvv) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [I2C0\_IRQHandler](#3cqmetx) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [I2C1\_IRQHandler](#1rvwp1q) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [LLWU\_IRQHandler](#4bvk7pj) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [LPTMR0\_IRQHandler](#2r0uhxc) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [LVD\_LVW\_IRQHandler](#1664s55) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [MCG\_IRQHandler](#3q5sasy) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [NMI\_Handler](#46r0co2) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [PIT\_IRQHandler](#25b2l0r) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [PORTA\_IRQHandler](#kgcv8k) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [PORTD\_IRQHandler](#34g0dwd) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [PendSV\_Handler](#3mzq4wv) from irq\_cm0.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [RTC\_IRQHandler](#1jlao46) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [RTC\_Seconds\_IRQHandler](#43ky6rz) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [Reserved20\_IRQHandler](#2iq8gzs) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [Reserved39\_IRQHandler](#xvir7l) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [Reserved45\_IRQHandler](#3hv69ve) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [Reset\_Handler](#1mrcu09) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [SPI0\_IRQHandler](#1x0gk37) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [SPI1\_IRQHandler](#4h042r0) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [SVC\_Handler](#1302m92) from irq\_cm0.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [SysTick\_Handler](#2250f4o) from irq\_cm0.o(.text) referenced from rtx\_kernel.o(.text.svcRtxKernelStart)
  + [SysTick\_Handler](#2250f4o) from irq\_cm0.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [SystemInit](#ymfzma) from system\_mkl25z4.o(i.SystemInit) referenced from startup\_mkl25z4.o(.text)
  + [TPM0\_IRQHandler](#2w5ecyt) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [TPM1\_IRQHandler](#1baon6m) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [TPM2\_IRQHandler](#3vac5uf) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [TSI0\_IRQHandler](#2afmg28) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [UART0\_IRQHandler](#pkwqa1) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [UART1\_IRQHandler](#39kk8xu) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [UART2\_IRQHandler](#1opuj5n) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [USB0\_IRQHandler](#48pi1tg) from startup\_mkl25z4.o(.text) referenced from startup\_mkl25z4.o(RESET)
  + [\_\_main](#gjdgxs) from \_\_main.o(!!!main) referenced from startup\_mkl25z4.o(.text)
  + [buttonThread](#2b6jogx) from main.o(i.buttonThread) referenced from main.o(i.main)
  + [ledsThread](#2olpkfy) from main.o(i.ledsThread) referenced from main.o(i.main)
  + [osRtxEventFlagsPostProcess](#1maplo9) from rtx\_evflags.o(.text.osRtxEventFlagsPostProcess) referenced from rtx\_evflags.o(.text.svcRtxEventFlagsNew)
  + [osRtxIdleThread](#2y3w247) from rtx\_config.o(i.osRtxIdleThread) referenced from rtx\_thread.o(.text.osRtxThreadStartup)
  + [osRtxMemoryInit](#38czs75) from rtx\_memory.o(.text.osRtxMemoryInit) referenced from rtx\_kernel.o(.text.svcRtxKernelInitialize)
  + [osRtxMessageQueuePostProcess](#1rf9gpq) from rtx\_msgqueue.o(.text.osRtxMessageQueuePostProcess) referenced from rtx\_msgqueue.o(.text.svcRtxMessageQueueNew)
  + [osRtxThreadPostProcess](#18vjpp8) from rtx\_thread.o(.text.osRtxThreadPostProcess) referenced from rtx\_thread.o(.text.svcRtxThreadNew)
  + [osRtxTimerThread](#356xmb2) from rtx\_timer.o(.text.osRtxTimerThread) referenced from rtx\_thread.o(.text.osRtxThreadStartup)
  + [osRtxTimerTick](#3xzr3ei) from rtx\_timer.o(.text.osRtxTimerTick) referenced from rtx\_timer.o(.text.osRtxTimerThread)
  + [osThreadExit](#1kc7wiv) from rtx\_thread.o(.text.osThreadExit) referenced from rtx\_thread.o(.text.svcRtxThreadNew)
  + [svcRtxDelay](#n5rssn) from rtx\_delay.o(.text.svcRtxDelay) referenced from rtx\_delay.o(.text.osDelay)
  + [svcRtxEventFlagsNew](#46ad4c2) from rtx\_evflags.o(.text.svcRtxEventFlagsNew) referenced from rtx\_evflags.o(.text.osEventFlagsNew)
  + [svcRtxEventFlagsSet](#2lfnejv) from rtx\_evflags.o(.text.svcRtxEventFlagsSet) referenced from rtx\_evflags.o(.text.osEventFlagsSet)
  + [svcRtxEventFlagsWait](#10kxoro) from rtx\_evflags.o(.text.svcRtxEventFlagsWait) referenced from rtx\_evflags.o(.text.osEventFlagsWait)
  + [svcRtxKernelGetState](#320vgez) from rtx\_kernel.o(.text.svcRtxKernelGetState) referenced from rtx\_kernel.o(.text.osKernelGetState)
  + [svcRtxKernelGetTickCount](#1h65qms) from rtx\_kernel.o(.text.svcRtxKernelGetTickCount) referenced from rtx\_kernel.o(.text.osKernelGetTickCount)
  + [svcRtxKernelInitialize](#415t9al) from rtx\_kernel.o(.text.svcRtxKernelInitialize) referenced from rtx\_kernel.o(.text.osKernelInitialize)
  + [svcRtxKernelStart](#2gb3jie) from rtx\_kernel.o(.text.svcRtxKernelStart) referenced from rtx\_kernel.o(.text.osKernelStart)
  + [svcRtxMessageQueueGet](#4bewzdj) from rtx\_msgqueue.o(.text.svcRtxMessageQueueGet) referenced from rtx\_msgqueue.o(.text.osMessageQueueGet)
  + [svcRtxMessageQueueNew](#2qk79lc) from rtx\_msgqueue.o(.text.svcRtxMessageQueueNew) referenced from rtx\_msgqueue.o(.text.osMessageQueueNew)
  + [svcRtxMessageQueuePut](#15phjt5) from rtx\_msgqueue.o(.text.svcRtxMessageQueuePut) referenced from rtx\_msgqueue.o(.text.osMessageQueuePut)
  + [svcRtxMutexAcquire](#3kkl7fh) from rtx\_mutex.o(.text.svcRtxMutexAcquire) referenced from rtx\_mutex.o(.text.osMutexAcquire)
  + [svcRtxMutexDelete](#1zpvhna) from rtx\_mutex.o(.text.svcRtxMutexDelete) referenced from rtx\_mutex.o(.text.osMutexDelete)
  + [svcRtxMutexNew](#4jpj0b3) from rtx\_mutex.o(.text.svcRtxMutexNew) referenced from rtx\_mutex.o(.text.osMutexNew)
  + [svcRtxMutexRelease](#2yutaiw) from rtx\_mutex.o(.text.svcRtxMutexRelease) referenced from rtx\_mutex.o(.text.osMutexRelease)
  + [svcRtxThreadExit](#3sv78d1) from rtx\_thread.o(.text.svcRtxThreadExit) referenced from rtx\_thread.o(.text.osThreadExit)
  + [svcRtxThreadNew](#280hiku) from rtx\_thread.o(.text.svcRtxThreadNew) referenced from rtx\_thread.o(.text.osThreadNew)

Global Symbols**\_\_main** (Thumb, 8 bytes, Stack size 0 bytes, \_\_main.o(!!!main))  
  
[Calls]

* + [>>](#1hmsyys)   \_\_rt\_entry
  + [>>](#30j0zll)   \_\_scatterload

**\_\_scatterload** (Thumb, 0 bytes, Stack size unknown bytes, \_\_scatter.o(!!!scatter))  
  
[Called By]

* + [>>](#gjdgxs)   \_\_main

**\_\_scatterload\_rt2** (Thumb, 52 bytes, Stack size unknown bytes, \_\_scatter.o(!!!scatter), UNUSED)  
  
[Calls]

* + [>>](#1hmsyys)   \_\_rt\_entry

**\_\_scatterload\_rt2\_thumb\_only** (Thumb, 0 bytes, Stack size unknown bytes, \_\_scatter.o(!!!scatter), UNUSED)**\_\_scatterload\_null** (Thumb, 0 bytes, Stack size unknown bytes, \_\_scatter.o(!!!scatter), UNUSED)**\_\_scatterload\_zeroinit** (Thumb, 28 bytes, Stack size unknown bytes, \_\_scatter\_zi.o(!!handler\_zi), UNUSED)**\_\_rt\_lib\_init** (Thumb, 0 bytes, Stack size unknown bytes, libinit.o(.ARM.Collect$$libinit$$00000000))  
  
[Called By]

* + [>>](#3fwokq0)   \_\_rt\_entry\_li

**\_\_rt\_lib\_init\_alloca\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$0000002E))**\_\_rt\_lib\_init\_argv\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$0000002C))**\_\_rt\_lib\_init\_atexit\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$0000001B))**\_\_rt\_lib\_init\_clock\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$00000021))**\_\_rt\_lib\_init\_cpp\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$00000032))**\_\_rt\_lib\_init\_exceptions\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$00000030))**\_\_rt\_lib\_init\_fp\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$00000002))**\_\_rt\_lib\_init\_fp\_trap\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$0000001F))**\_\_rt\_lib\_init\_getenv\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$00000023))**\_\_rt\_lib\_init\_heap\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$0000000A))**\_\_rt\_lib\_init\_lc\_collate\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$00000011))**\_\_rt\_lib\_init\_lc\_ctype\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$00000013))**\_\_rt\_lib\_init\_lc\_monetary\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$00000015))**\_\_rt\_lib\_init\_lc\_numeric\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$00000017))**\_\_rt\_lib\_init\_lc\_time\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$00000019))**\_\_rt\_lib\_init\_preinit\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$00000004))**\_\_rt\_lib\_init\_rand\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$0000000E))**\_\_rt\_lib\_init\_return** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$00000033))**\_\_rt\_lib\_init\_signal\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$0000001D))**\_\_rt\_lib\_init\_stdio\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$00000025))**\_\_rt\_lib\_init\_user\_alloc\_1** (Thumb, 0 bytes, Stack size unknown bytes, libinit2.o(.ARM.Collect$$libinit$$0000000C))**\_\_rt\_lib\_shutdown** (Thumb, 0 bytes, Stack size unknown bytes, libshutdown.o(.ARM.Collect$$libshutdown$$00000000))  
  
[Called By]

* + [>>](#19c6y18)   \_\_rt\_exit\_ls

**\_\_rt\_lib\_shutdown\_cpp\_1** (Thumb, 0 bytes, Stack size unknown bytes, libshutdown2.o(.ARM.Collect$$libshutdown$$00000002))**\_\_rt\_lib\_shutdown\_fp\_trap\_1** (Thumb, 0 bytes, Stack size unknown bytes, libshutdown2.o(.ARM.Collect$$libshutdown$$00000007))**\_\_rt\_lib\_shutdown\_heap\_1** (Thumb, 0 bytes, Stack size unknown bytes, libshutdown2.o(.ARM.Collect$$libshutdown$$0000000F))**\_\_rt\_lib\_shutdown\_return** (Thumb, 0 bytes, Stack size unknown bytes, libshutdown2.o(.ARM.Collect$$libshutdown$$00000010))**\_\_rt\_lib\_shutdown\_signal\_1** (Thumb, 0 bytes, Stack size unknown bytes, libshutdown2.o(.ARM.Collect$$libshutdown$$0000000A))**\_\_rt\_lib\_shutdown\_stdio\_1** (Thumb, 0 bytes, Stack size unknown bytes, libshutdown2.o(.ARM.Collect$$libshutdown$$00000004))**\_\_rt\_lib\_shutdown\_user\_alloc\_1** (Thumb, 0 bytes, Stack size unknown bytes, libshutdown2.o(.ARM.Collect$$libshutdown$$0000000C))**\_\_rt\_entry** (Thumb, 0 bytes, Stack size unknown bytes, \_\_rtentry.o(.ARM.Collect$$rtentry$$00000000))  
  
[Called By]

* + [>>](#gjdgxs)   \_\_main
  + [>>](#1fob9te)   \_\_scatterload\_rt2

**\_\_rt\_entry\_presh\_1** (Thumb, 0 bytes, Stack size unknown bytes, \_\_rtentry2.o(.ARM.Collect$$rtentry$$00000002))**\_\_rt\_entry\_sh** (Thumb, 0 bytes, Stack size unknown bytes, \_\_rtentry4.o(.ARM.Collect$$rtentry$$00000004))  
  
[Stack]

* + Max Depth = 8 + Unknown Stack Size
  + Call Chain = \_\_rt\_entry\_sh ⇒ \_\_user\_setup\_stackheap

[Calls]

* + [>>](#279ka65)   \_\_user\_setup\_stackheap

**\_\_rt\_entry\_postsh\_1** (Thumb, 0 bytes, Stack size unknown bytes, \_\_rtentry2.o(.ARM.Collect$$rtentry$$00000008))  
  
[Stack]

* + Max Depth = 16 + Unknown Stack Size
  + Call Chain = \_\_rt\_entry\_postsh\_1 ⇒ \_platform\_post\_stackheap\_init ⇒ osKernelInitialize

[Calls]

* + [>>](#3w19e94)   \_platform\_post\_stackheap\_init

**\_\_rt\_entry\_li** (Thumb, 0 bytes, Stack size unknown bytes, \_\_rtentry2.o(.ARM.Collect$$rtentry$$0000000A))  
  
[Calls]

* + [>>](#3dy6vkm)   \_\_rt\_lib\_init

**\_\_rt\_entry\_main** (Thumb, 0 bytes, Stack size unknown bytes, \_\_rtentry2.o(.ARM.Collect$$rtentry$$0000000D))  
  
[Stack]

* + Max Depth = 24 + Unknown Stack Size
  + Call Chain = \_\_rt\_entry\_main ⇒ main ⇒ SystemCoreClockUpdate ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#13qzunr)   main
  + [>>](#meukdy)   exit

**\_\_rt\_entry\_postli\_1** (Thumb, 0 bytes, Stack size unknown bytes, \_\_rtentry2.o(.ARM.Collect$$rtentry$$0000000C))**\_\_rt\_exit** (Thumb, 0 bytes, Stack size unknown bytes, rtexit.o(.ARM.Collect$$rtexit$$00000000))  
  
[Called By]

* + [>>](#meukdy)   exit

**\_\_rt\_exit\_ls** (Thumb, 0 bytes, Stack size unknown bytes, rtexit2.o(.ARM.Collect$$rtexit$$00000003))  
  
[Calls]

* + [>>](#1pxezwc)   \_\_rt\_lib\_shutdown

**\_\_rt\_exit\_prels\_1** (Thumb, 0 bytes, Stack size unknown bytes, rtexit2.o(.ARM.Collect$$rtexit$$00000002))**\_\_rt\_exit\_exit** (Thumb, 0 bytes, Stack size unknown bytes, rtexit2.o(.ARM.Collect$$rtexit$$00000004))  
  
[Calls]

* + [>>](#45jfvxd)   \_sys\_exit

**\_\_aeabi\_memcpy4** (Thumb, 56 bytes, Stack size 0 bytes, rt\_memcpy.o(.emb\_text))  
  
[Called By]

* + [>>](#haapch)   \_\_aeabi\_memcpy

**\_\_aeabi\_memcpy8** (Thumb, 0 bytes, Stack size 0 bytes, rt\_memcpy.o(.emb\_text), UNUSED)**Reset\_Handler** (Thumb, 12 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))**NMI\_Handler** (Thumb, 2 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
  
[Calls]

* + [>>](#46r0co2)   NMI\_Handler

[Called By]

* + [>>](#46r0co2)   NMI\_Handler

[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**HardFault\_Handler** (Thumb, 2 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
  
[Calls]

* + [>>](#2lwamvv)   HardFault\_Handler

[Called By]

* + [>>](#2lwamvv)   HardFault\_Handler

[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**ADC0\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**CMP0\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**DAC0\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**DMA0\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**DMA1\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**DMA2\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**DMA3\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**DefaultISR** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(.text)

**FTFA\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**I2C0\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**I2C1\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**LLWU\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**LPTMR0\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**LVD\_LVW\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**MCG\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**PIT\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**PORTA\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**PORTD\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**RTC\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**RTC\_Seconds\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**Reserved20\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**Reserved39\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**Reserved45\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**SPI0\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**SPI1\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**TPM0\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**TPM1\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**TPM2\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**TSI0\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**UART0\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**UART1\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**UART2\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**USB0\_IRQHandler** (Thumb, 0 bytes, Stack size 0 bytes, startup\_mkl25z4.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**\_\_user\_initial\_stackheap** (Thumb, 0 bytes, Stack size unknown bytes, startup\_mkl25z4.o(.text))  
  
[Called By]

* + [>>](#279ka65)   \_\_user\_setup\_stackheap

**SVC\_Handler** (Thumb, 132 bytes, Stack size 0 bytes, irq\_cm0.o(.text))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**PendSV\_Handler** (Thumb, 12 bytes, Stack size 0 bytes, irq\_cm0.o(.text))  
  
[Stack]

* + Max Depth = 64
  + Call Chain = PendSV\_Handler ⇒ osRtxPendSV\_Handler ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#3ls5o66)   osRtxPendSV\_Handler

[Address Reference Count : 1]

* + startup\_mkl25z4.o(RESET)

**SysTick\_Handler** (Thumb, 12 bytes, Stack size 0 bytes, irq\_cm0.o(.text))  
  
[Stack]

* + Max Depth = 56
  + Call Chain = SysTick\_Handler ⇒ osRtxTick\_Handler ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#l7a3n9)   osRtxTick\_Handler

[Address Reference Count : 2]

* + rtx\_kernel.o(.text.svcRtxKernelStart)
  + startup\_mkl25z4.o(RESET)

**\_\_aeabi\_memcpy** (Thumb, 130 bytes, Stack size 24 bytes, rt\_memcpy.o(.text))  
  
[Stack]

* + Max Depth = 24
  + Call Chain = \_\_aeabi\_memcpy

[Calls]

* + [>>](#nmf14n)   \_\_aeabi\_memcpy4

[Called By]

* + [>>](#1vsw3ci)   osMessageQueuePut
  + [>>](#wnyagw)   osMessageQueueGet
  + [>>](#15phjt5)   svcRtxMessageQueuePut
  + [>>](#4bewzdj)   svcRtxMessageQueueGet
  + [>>](#1rf9gpq)   osRtxMessageQueuePostProcess

**\_\_rt\_memcpy** (Thumb, 0 bytes, Stack size 24 bytes, rt\_memcpy.o(.text), UNUSED)**\_memset\_w** (Thumb, 26 bytes, Stack size 0 bytes, rt\_memclr.o(.text))  
  
[Called By]

* + [>>](#4du1wux)   \_\_rt\_memclr\_w
  + [>>](#40ew0vw)   \_memset

**\_memset** (Thumb, 30 bytes, Stack size 0 bytes, rt\_memclr.o(.text))  
  
[Calls]

* + [>>](#1gf8i83)   \_memset\_w

[Called By]

* + [>>](#2fk6b3p)   \_\_aeabi\_memclr

**\_\_aeabi\_memclr** (Thumb, 4 bytes, Stack size 0 bytes, rt\_memclr.o(.text))  
  
[Calls]

* + [>>](#40ew0vw)   \_memset

[Called By]

* + [>>](#280hiku)   svcRtxThreadNew
  + [>>](#2qk79lc)   svcRtxMessageQueueNew

**\_\_rt\_memclr** (Thumb, 0 bytes, Stack size 0 bytes, rt\_memclr.o(.text), UNUSED)**\_\_aeabi\_memclr4** (Thumb, 0 bytes, Stack size 0 bytes, rt\_memclr.o(.text))  
  
[Called By]

* + [>>](#415t9al)   svcRtxKernelInitialize
  + [>>](#4jpj0b3)   svcRtxMutexNew
  + [>>](#280hiku)   svcRtxThreadNew

**\_\_aeabi\_memclr8** (Thumb, 0 bytes, Stack size 0 bytes, rt\_memclr.o(.text), UNUSED)**\_\_rt\_memclr\_w** (Thumb, 4 bytes, Stack size 0 bytes, rt\_memclr.o(.text), UNUSED)  
  
[Calls]

* + [>>](#1gf8i83)   \_memset\_w

**\_\_use\_two\_region\_memory** (Thumb, 2 bytes, Stack size 0 bytes, heapauxi.o(.text), UNUSED)**\_\_rt\_heap\_escrow$2region** (Thumb, 2 bytes, Stack size 0 bytes, heapauxi.o(.text), UNUSED)**\_\_rt\_heap\_expand$2region** (Thumb, 2 bytes, Stack size 0 bytes, heapauxi.o(.text), UNUSED)**\_\_user\_setup\_stackheap** (Thumb, 62 bytes, Stack size 8 bytes, sys\_stackheap\_outer.o(.text))  
  
[Stack]

* + Max Depth = 8 + Unknown Stack Size
  + Call Chain = \_\_user\_setup\_stackheap

[Calls]

* + [>>](#2nusc19)   \_\_user\_initial\_stackheap
  + [>>](#1ljsd9k)   \_\_user\_perproc\_libspace

[Called By]

* + [>>](#2grqrue)   \_\_rt\_entry\_sh

**exit** (Thumb, 16 bytes, Stack size 8 bytes, exit.o(.text))  
  
[Stack]

* + Max Depth = 8 + Unknown Stack Size
  + Call Chain = exit

[Calls]

* + [>>](#2u6wntf)   \_\_rt\_exit
  + [>>](#3pp52gy)   \_call\_atexit\_fns (Weak Reference)

[Called By]

* + [>>](#1v1yuxt)   \_\_rt\_entry\_main

**\_\_user\_libspace** (Thumb, 8 bytes, Stack size 0 bytes, libspace.o(.text), UNUSED)**\_\_user\_perproc\_libspace** (Thumb, 0 bytes, Stack size 0 bytes, libspace.o(.text))  
  
[Called By]

* + [>>](#279ka65)   \_\_user\_setup\_stackheap

**\_sys\_exit** (Thumb, 8 bytes, Stack size 0 bytes, sys\_exit.o(.text))  
  
[Called By]

* + [>>](#28h4qwu)   \_\_rt\_exit\_exit

**\_\_I$use$semihosting** (Thumb, 0 bytes, Stack size 0 bytes, use\_no\_semi.o(.text), UNUSED)**\_\_use\_no\_semihosting\_swi** (Thumb, 2 bytes, Stack size 0 bytes, use\_no\_semi.o(.text), UNUSED)**\_\_decompress** (Thumb, 0 bytes, Stack size unknown bytes, \_\_dczerorl.o(.text), UNUSED)**\_\_decompress0** (Thumb, 58 bytes, Stack size unknown bytes, \_\_dczerorl.o(.text), UNUSED)**\_\_semihosting\_library\_function** (Thumb, 0 bytes, Stack size unknown bytes, indicate\_semi.o(.text), UNUSED)**osRtxIdleThread** (Thumb, 4 bytes, Stack size 0 bytes, rtx\_config.o(i.osRtxIdleThread))  
[Address Reference Count : 1]

* + rtx\_thread.o(.text.osRtxThreadStartup)

**\_\_aeabi\_uidiv** (Thumb, 0 bytes, Stack size 0 bytes, aeabi\_sdiv.o(.text))  
  
[Called By]

* + [>>](#2pta16n)   OS\_Tick\_Setup

**\_\_aeabi\_uidivmod** (Thumb, 20 bytes, Stack size 0 bytes, aeabi\_sdiv.o(.text))  
  
[Called By]

* + [>>](#2jh5peh)   SystemCoreClockUpdate

**\_\_aeabi\_idiv** (Thumb, 0 bytes, Stack size 8 bytes, aeabi\_sdiv.o(.text), UNUSED)**\_\_aeabi\_idivmod** (Thumb, 326 bytes, Stack size 8 bytes, aeabi\_sdiv.o(.text), UNUSED)**OS\_Tick\_AcknowledgeIRQ** (Thumb, 12 bytes, Stack size 0 bytes, os\_systick.o(.text.OS\_Tick\_AcknowledgeIRQ))  
  
[Called By]

* + [>>](#l7a3n9)   osRtxTick\_Handler

**OS\_Tick\_Enable** (Thumb, 44 bytes, Stack size 0 bytes, os\_systick.o(.text.OS\_Tick\_Enable))  
  
[Called By]

* + [>>](#2gb3jie)   svcRtxKernelStart

**OS\_Tick\_GetIRQn** (Thumb, 6 bytes, Stack size 0 bytes, os\_systick.o(.text.OS\_Tick\_GetIRQn))  
  
[Called By]

* + [>>](#2gb3jie)   svcRtxKernelStart

**OS\_Tick\_Setup** (Thumb, 76 bytes, Stack size 8 bytes, os\_systick.o(.text.OS\_Tick\_Setup))  
  
[Stack]

* + Max Depth = 8
  + Call Chain = OS\_Tick\_Setup

[Calls]

* + [>>](#1d96cc0)   \_\_aeabi\_uidiv

[Called By]

* + [>>](#2gb3jie)   svcRtxKernelStart

**osDelay** (Thumb, 40 bytes, Stack size 8 bytes, rtx\_delay.o(.text.osDelay))  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osDelay

[Called By]

* + [>>](#2b6jogx)   buttonThread

**osEventFlagsNew** (Thumb, 36 bytes, Stack size 8 bytes, rtx\_evflags.o(.text.osEventFlagsNew))  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osEventFlagsNew

[Called By]

* + [>>](#13qzunr)   main

**osEventFlagsSet** (Thumb, 88 bytes, Stack size 16 bytes, rtx\_evflags.o(.text.osEventFlagsSet))  
  
[Stack]

* + Max Depth = 48
  + Call Chain = osEventFlagsSet ⇒ osRtxPostProcess ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#20xfydz)   osRtxPostProcess

[Called By]

* + [>>](#2b6jogx)   buttonThread

**osEventFlagsWait** (Thumb, 80 bytes, Stack size 16 bytes, rtx\_evflags.o(.text.osEventFlagsWait))  
  
[Stack]

* + Max Depth = 32
  + Call Chain = osEventFlagsWait ⇒ EventFlagsCheck

[Calls]

* + [>>](#375fbgg)   EventFlagsCheck

[Called By]

* + [>>](#2olpkfy)   ledsThread

**osKernelGetState** (Thumb, 48 bytes, Stack size 8 bytes, rtx\_kernel.o(.text.osKernelGetState), UNUSED, KEPT)  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osKernelGetState

[Called By]

* + [>>](#i17xr6)   os\_kernel\_is\_active

**osKernelGetTickCount** (Thumb, 40 bytes, Stack size 8 bytes, rtx\_kernel.o(.text.osKernelGetTickCount))  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osKernelGetTickCount

[Called By]

* + [>>](#2olpkfy)   ledsThread

**osKernelInitialize** (Thumb, 36 bytes, Stack size 8 bytes, rtx\_kernel.o(.text.osKernelInitialize))  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osKernelInitialize

[Called By]

* + [>>](#3w19e94)   \_platform\_post\_stackheap\_init
  + [>>](#13qzunr)   main

**osKernelStart** (Thumb, 36 bytes, Stack size 8 bytes, rtx\_kernel.o(.text.osKernelStart))  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osKernelStart

[Called By]

* + [>>](#13qzunr)   main

**osMessageQueueGet** (Thumb, 120 bytes, Stack size 24 bytes, rtx\_msgqueue.o(.text.osMessageQueueGet))  
  
[Stack]

* + Max Depth = 56
  + Call Chain = osMessageQueueGet ⇒ osRtxPostProcess ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#20xfydz)   osRtxPostProcess
  + [>>](#haapch)   \_\_aeabi\_memcpy
  + [>>](#2d51dmb)   MessageQueueGet

[Called By]

* + [>>](#356xmb2)   osRtxTimerThread

**osMessageQueueNew** (Thumb, 36 bytes, Stack size 16 bytes, rtx\_msgqueue.o(.text.osMessageQueueNew))  
  
[Stack]

* + Max Depth = 16
  + Call Chain = osMessageQueueNew

[Called By]

* + [>>](#356xmb2)   osRtxTimerThread

**osMessageQueuePut** (Thumb, 128 bytes, Stack size 24 bytes, rtx\_msgqueue.o(.text.osMessageQueuePut))  
  
[Stack]

* + Max Depth = 56
  + Call Chain = osMessageQueuePut ⇒ osRtxPostProcess ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#20xfydz)   osRtxPostProcess
  + [>>](#1nia2ey)   osRtxMemoryPoolAlloc
  + [>>](#haapch)   \_\_aeabi\_memcpy

[Called By]

* + [>>](#3xzr3ei)   osRtxTimerTick

**osMutexAcquire** (Thumb, 40 bytes, Stack size 8 bytes, rtx\_mutex.o(.text.osMutexAcquire), UNUSED, KEPT)  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osMutexAcquire

[Called By]

* + [>>](#1xrdshw)   \_mutex\_acquire

**osMutexDelete** (Thumb, 40 bytes, Stack size 8 bytes, rtx\_mutex.o(.text.osMutexDelete), UNUSED, KEPT)  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osMutexDelete

[Called By]

* + [>>](#4hr1b5p)   \_mutex\_free

**osMutexNew** (Thumb, 36 bytes, Stack size 8 bytes, rtx\_mutex.o(.text.osMutexNew), UNUSED, KEPT)  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osMutexNew

[Called By]

* + [>>](#2wwbldi)   \_mutex\_initialize

**osMutexRelease** (Thumb, 40 bytes, Stack size 8 bytes, rtx\_mutex.o(.text.osMutexRelease), UNUSED, KEPT)  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osMutexRelease

[Called By]

* + [>>](#1c1lvlb)   \_mutex\_release

**osRtxMemoryAlloc** (Thumb, 124 bytes, Stack size 28 bytes, rtx\_memory.o(.text.osRtxMemoryAlloc))  
  
[Stack]

* + Max Depth = 28
  + Call Chain = osRtxMemoryAlloc

[Called By]

* + [>>](#4jpj0b3)   svcRtxMutexNew
  + [>>](#46ad4c2)   svcRtxEventFlagsNew
  + [>>](#280hiku)   svcRtxThreadNew
  + [>>](#2qk79lc)   svcRtxMessageQueueNew

**osRtxMemoryFree** (Thumb, 78 bytes, Stack size 20 bytes, rtx\_memory.o(.text.osRtxMemoryFree))  
  
[Stack]

* + Max Depth = 20
  + Call Chain = osRtxMemoryFree

[Called By]

* + [>>](#1zpvhna)   svcRtxMutexDelete
  + [>>](#280hiku)   svcRtxThreadNew
  + [>>](#2tq9fhf)   osRtxThreadFree
  + [>>](#2qk79lc)   svcRtxMessageQueueNew

**osRtxMemoryInit** (Thumb, 58 bytes, Stack size 8 bytes, rtx\_memory.o(.text.osRtxMemoryInit))  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osRtxMemoryInit

[Called By]

* + [>>](#415t9al)   svcRtxKernelInitialize

[Address Reference Count : 1]

* + rtx\_kernel.o(.text.svcRtxKernelInitialize)

**osRtxMemoryPoolAlloc** (Thumb, 56 bytes, Stack size 8 bytes, rtx\_mempool.o(.text.osRtxMemoryPoolAlloc))  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osRtxMemoryPoolAlloc

[Called By]

* + [>>](#1vsw3ci)   osMessageQueuePut
  + [>>](#4jpj0b3)   svcRtxMutexNew
  + [>>](#46ad4c2)   svcRtxEventFlagsNew
  + [>>](#280hiku)   svcRtxThreadNew
  + [>>](#15phjt5)   svcRtxMessageQueuePut
  + [>>](#2qk79lc)   svcRtxMessageQueueNew
  + [>>](#4bewzdj)   svcRtxMessageQueueGet
  + [>>](#1rf9gpq)   osRtxMessageQueuePostProcess

**osRtxMemoryPoolFree** (Thumb, 64 bytes, Stack size 8 bytes, rtx\_mempool.o(.text.osRtxMemoryPoolFree))  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osRtxMemoryPoolFree

[Called By]

* + [>>](#1zpvhna)   svcRtxMutexDelete
  + [>>](#280hiku)   svcRtxThreadNew
  + [>>](#2tq9fhf)   osRtxThreadFree
  + [>>](#2qk79lc)   svcRtxMessageQueueNew
  + [>>](#4bewzdj)   svcRtxMessageQueueGet
  + [>>](#1rf9gpq)   osRtxMessageQueuePostProcess

**osRtxMemoryPoolInit** (Thumb, 68 bytes, Stack size 16 bytes, rtx\_mempool.o(.text.osRtxMemoryPoolInit))  
  
[Stack]

* + Max Depth = 16
  + Call Chain = osRtxMemoryPoolInit

[Called By]

* + [>>](#415t9al)   svcRtxKernelInitialize
  + [>>](#2qk79lc)   svcRtxMessageQueueNew

**osRtxMutexOwnerRelease** (Thumb, 62 bytes, Stack size 24 bytes, rtx\_mutex.o(.text.osRtxMutexOwnerRelease))  
  
[Stack]

* + Max Depth = 80
  + Call Chain = osRtxMutexOwnerRelease ⇒ osRtxThreadWaitExit ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#1f7o1he)   osRtxThreadListGet
  + [>>](#261ztfg)   osRtxThreadWaitExit

[Called By]

* + [>>](#3sv78d1)   svcRtxThreadExit

**osRtxPendSV\_Handler** (Thumb, 136 bytes, Stack size 24 bytes, rtx\_system.o(.text.osRtxPendSV\_Handler))  
  
[Stack]

* + Max Depth = 64
  + Call Chain = osRtxPendSV\_Handler ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#302dr9l)   osRtxThreadDispatch

[Called By]

* + [>>](#3mzq4wv)   PendSV\_Handler

**osRtxPostProcess** (Thumb, 108 bytes, Stack size 24 bytes, rtx\_system.o(.text.osRtxPostProcess))  
  
[Stack]

* + Max Depth = 32
  + Call Chain = osRtxPostProcess ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#3nqndbk)   osRtxErrorNotify

[Called By]

* + [>>](#243i4a2)   osEventFlagsSet
  + [>>](#1vsw3ci)   osMessageQueuePut
  + [>>](#wnyagw)   osMessageQueueGet

**osRtxThreadDelayTick** (Thumb, 60 bytes, Stack size 16 bytes, rtx\_thread.o(.text.osRtxThreadDelayTick))  
  
[Stack]

* + Max Depth = 40
  + Call Chain = osRtxThreadDelayTick ⇒ osRtxThreadReadyPut ⇒ osRtxThreadListPut

[Calls]

* + [>>](#3dhjn8m)   osRtxThreadReadyPut
  + [>>](#2eclud0)   osRtxThreadListRemove

[Called By]

* + [>>](#l7a3n9)   osRtxTick\_Handler

**osRtxThreadDispatch** (Thumb, 88 bytes, Stack size 16 bytes, rtx\_thread.o(.text.osRtxThreadDispatch))  
  
[Stack]

* + Max Depth = 40
  + Call Chain = osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#16x20ju)   osRtxThreadSwitch
  + [>>](#3dhjn8m)   osRtxThreadReadyPut
  + [>>](#2eclud0)   osRtxThreadListRemove
  + [>>](#3fg1ce0)   osRtxThreadBlock

[Called By]

* + [>>](#l7a3n9)   osRtxTick\_Handler
  + [>>](#3ls5o66)   osRtxPendSV\_Handler
  + [>>](#2yutaiw)   svcRtxMutexRelease
  + [>>](#1zpvhna)   svcRtxMutexDelete
  + [>>](#2lfnejv)   svcRtxEventFlagsSet
  + [>>](#261ztfg)   osRtxThreadWaitExit
  + [>>](#280hiku)   svcRtxThreadNew

**osRtxThreadListGet** (Thumb, 24 bytes, Stack size 0 bytes, rtx\_thread.o(.text.osRtxThreadListGet))  
  
[Called By]

* + [>>](#2gb3jie)   svcRtxKernelStart
  + [>>](#2yutaiw)   svcRtxMutexRelease
  + [>>](#1zpvhna)   svcRtxMutexDelete
  + [>>](#3qwpj7n)   osRtxThreadWaitEnter
  + [>>](#11si5id)   osRtxMutexOwnerRelease
  + [>>](#3sv78d1)   svcRtxThreadExit
  + [>>](#15phjt5)   svcRtxMessageQueuePut
  + [>>](#4bewzdj)   svcRtxMessageQueueGet
  + [>>](#1rf9gpq)   osRtxMessageQueuePostProcess

**osRtxThreadListPut** (Thumb, 58 bytes, Stack size 16 bytes, rtx\_thread.o(.text.osRtxThreadListPut))  
  
[Stack]

* + Max Depth = 16
  + Call Chain = osRtxThreadListPut

[Called By]

* + [>>](#3kkl7fh)   svcRtxMutexAcquire
  + [>>](#10kxoro)   svcRtxEventFlagsWait
  + [>>](#3dhjn8m)   osRtxThreadReadyPut
  + [>>](#thw4kt)   osRtxThreadListSort
  + [>>](#15phjt5)   svcRtxMessageQueuePut
  + [>>](#4bewzdj)   svcRtxMessageQueueGet

**osRtxThreadListRemove** (Thumb, 24 bytes, Stack size 0 bytes, rtx\_thread.o(.text.osRtxThreadListRemove))  
  
[Called By]

* + [>>](#302dr9l)   osRtxThreadDispatch
  + [>>](#4kx3h1s)   osRtxThreadDelayTick
  + [>>](#l7a3n9)   osRtxTick\_Handler
  + [>>](#2lfnejv)   svcRtxEventFlagsSet
  + [>>](#1maplo9)   osRtxEventFlagsPostProcess
  + [>>](#thw4kt)   osRtxThreadListSort

**osRtxThreadListSort** (Thumb, 38 bytes, Stack size 16 bytes, rtx\_thread.o(.text.osRtxThreadListSort))  
  
[Stack]

* + Max Depth = 32
  + Call Chain = osRtxThreadListSort ⇒ osRtxThreadListPut

[Calls]

* + [>>](#2eclud0)   osRtxThreadListRemove
  + [>>](#3z7bk57)   osRtxThreadListPut

[Called By]

* + [>>](#1zpvhna)   svcRtxMutexDelete
  + [>>](#3kkl7fh)   svcRtxMutexAcquire

**osRtxThreadReadyPut** (Thumb, 24 bytes, Stack size 8 bytes, rtx\_thread.o(.text.osRtxThreadReadyPut))  
  
[Stack]

* + Max Depth = 24
  + Call Chain = osRtxThreadReadyPut ⇒ osRtxThreadListPut

[Calls]

* + [>>](#3z7bk57)   osRtxThreadListPut

[Called By]

* + [>>](#302dr9l)   osRtxThreadDispatch
  + [>>](#4kx3h1s)   osRtxThreadDelayTick
  + [>>](#l7a3n9)   osRtxTick\_Handler
  + [>>](#261ztfg)   osRtxThreadWaitExit

**osRtxThreadRegPtr** (Thumb, 6 bytes, Stack size 0 bytes, rtx\_thread.o(.text.osRtxThreadRegPtr))  
  
[Called By]

* + [>>](#15phjt5)   svcRtxMessageQueuePut
  + [>>](#4bewzdj)   svcRtxMessageQueueGet
  + [>>](#1rf9gpq)   osRtxMessageQueuePostProcess

**osRtxThreadStackCheck** (Thumb, 44 bytes, Stack size 8 bytes, rtx\_thread.o(.text.osRtxThreadStackCheck))  
  
[Stack]

* + Max Depth = 16
  + Call Chain = osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#3nqndbk)   osRtxErrorNotify

[Called By]

* + [>>](#16x20ju)   osRtxThreadSwitch

**osRtxThreadStartup** (Thumb, 92 bytes, Stack size 16 bytes, rtx\_thread.o(.text.osRtxThreadStartup))  
  
[Stack]

* + Max Depth = 104
  + Call Chain = osRtxThreadStartup ⇒ svcRtxThreadNew ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#280hiku)   svcRtxThreadNew

[Called By]

* + [>>](#2gb3jie)   svcRtxKernelStart

**osRtxThreadSwitch** (Thumb, 20 bytes, Stack size 8 bytes, rtx\_thread.o(.text.osRtxThreadSwitch))  
  
[Stack]

* + Max Depth = 24
  + Call Chain = osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#4cmhg48)   osRtxThreadStackCheck

[Called By]

* + [>>](#302dr9l)   osRtxThreadDispatch
  + [>>](#2gb3jie)   svcRtxKernelStart
  + [>>](#l7a3n9)   osRtxTick\_Handler
  + [>>](#3qwpj7n)   osRtxThreadWaitEnter
  + [>>](#3sv78d1)   svcRtxThreadExit

**osRtxThreadWaitEnter** (Thumb, 56 bytes, Stack size 16 bytes, rtx\_thread.o(.text.osRtxThreadWaitEnter))  
  
[Stack]

* + Max Depth = 40
  + Call Chain = osRtxThreadWaitEnter ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#16x20ju)   osRtxThreadSwitch
  + [>>](#1f7o1he)   osRtxThreadListGet
  + [>>](#1ulbmlt)   osRtxThreadDelayInsert

[Called By]

* + [>>](#3kkl7fh)   svcRtxMutexAcquire
  + [>>](#10kxoro)   svcRtxEventFlagsWait
  + [>>](#n5rssn)   svcRtxDelay
  + [>>](#15phjt5)   svcRtxMessageQueuePut
  + [>>](#4bewzdj)   svcRtxMessageQueueGet

**osRtxThreadWaitExit** (Thumb, 34 bytes, Stack size 16 bytes, rtx\_thread.o(.text.osRtxThreadWaitExit))  
  
[Stack]

* + Max Depth = 56
  + Call Chain = osRtxThreadWaitExit ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#302dr9l)   osRtxThreadDispatch
  + [>>](#3dhjn8m)   osRtxThreadReadyPut
  + [>>](#4ekz59m)   osRtxThreadDelayRemove

[Called By]

* + [>>](#2yutaiw)   svcRtxMutexRelease
  + [>>](#1zpvhna)   svcRtxMutexDelete
  + [>>](#2lfnejv)   svcRtxEventFlagsSet
  + [>>](#1maplo9)   osRtxEventFlagsPostProcess
  + [>>](#11si5id)   osRtxMutexOwnerRelease
  + [>>](#3sv78d1)   svcRtxThreadExit
  + [>>](#18vjpp8)   osRtxThreadPostProcess
  + [>>](#15phjt5)   svcRtxMessageQueuePut
  + [>>](#4bewzdj)   svcRtxMessageQueueGet
  + [>>](#1rf9gpq)   osRtxMessageQueuePostProcess

**osRtxTick\_Handler** (Thumb, 116 bytes, Stack size 16 bytes, rtx\_system.o(.text.osRtxTick\_Handler))  
  
[Stack]

* + Max Depth = 56
  + Call Chain = osRtxTick\_Handler ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#16x20ju)   osRtxThreadSwitch
  + [>>](#302dr9l)   osRtxThreadDispatch
  + [>>](#4kx3h1s)   osRtxThreadDelayTick
  + [>>](#3dhjn8m)   osRtxThreadReadyPut
  + [>>](#2eclud0)   osRtxThreadListRemove
  + [>>](#3bj1y38)   OS\_Tick\_AcknowledgeIRQ

[Called By]

* + [>>](#2250f4o)   SysTick\_Handler

**osRtxTimerThread** (Thumb, 64 bytes, Stack size 16 bytes, rtx\_timer.o(.text.osRtxTimerThread))  
  
[Stack]

* + Max Depth = 72
  + Call Chain = osRtxTimerThread ⇒ osMessageQueueGet ⇒ osRtxPostProcess ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#3gnlt4p)   osMessageQueueNew
  + [>>](#wnyagw)   osMessageQueueGet

[Address Reference Count : 1]

* + rtx\_thread.o(.text.osRtxThreadStartup)

**osThreadExit** (Thumb, 12 bytes, Stack size 8 bytes, rtx\_thread.o(.text.osThreadExit))  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osThreadExit

[Address Reference Count : 1]

* + rtx\_thread.o(.text.svcRtxThreadNew)

**osThreadNew** (Thumb, 36 bytes, Stack size 16 bytes, rtx\_thread.o(.text.osThreadNew))  
  
[Stack]

* + Max Depth = 16
  + Call Chain = osThreadNew

[Called By]

* + [>>](#13qzunr)   main

**SystemCoreClockUpdate** (Thumb, 372 bytes, Stack size 16 bytes, system\_mkl25z4.o(i.SystemCoreClockUpdate))  
  
[Stack]

* + Max Depth = 24
  + Call Chain = SystemCoreClockUpdate ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#3im3ia3)   \_\_ARM\_common\_switch8
  + [>>](#3x8tuzt)   \_\_aeabi\_uidivmod

[Called By]

* + [>>](#13qzunr)   main

**SystemInit** (Thumb, 8 bytes, Stack size 0 bytes, system\_mkl25z4.o(i.SystemInit))  
[Address Reference Count : 1]

* + startup\_mkl25z4.o(.text)

**\_\_ARM\_common\_switch8** (Thumb, 28 bytes, Stack size 8 bytes, rtx\_config.o(i.\_\_ARM\_common\_switch8))  
  
[Stack]

* + Max Depth = 8
  + Call Chain = \_\_ARM\_common\_switch8

[Called By]

* + [>>](#3nqndbk)   osRtxErrorNotify
  + [>>](#2jh5peh)   SystemCoreClockUpdate

**\_mutex\_acquire** (Thumb, 24 bytes, Stack size 8 bytes, rtx\_lib.o(i.\_mutex\_acquire), UNUSED, KEPT)  
  
[Stack]

* + Max Depth = 24
  + Call Chain = \_mutex\_acquire ⇒ os\_kernel\_is\_active ⇒ osKernelGetState

[Calls]

* + [>>](#4fsjm0b)   osMutexAcquire
  + [>>](#i17xr6)   os\_kernel\_is\_active

**\_mutex\_free** (Thumb, 12 bytes, Stack size 8 bytes, rtx\_lib.o(i.\_mutex\_free), UNUSED, KEPT)  
  
[Stack]

* + Max Depth = 16
  + Call Chain = \_mutex\_free ⇒ osMutexDelete

[Calls]

* + [>>](#2uxtw84)   osMutexDelete

**\_mutex\_initialize** (Thumb, 36 bytes, Stack size 16 bytes, rtx\_lib.o(i.\_mutex\_initialize), UNUSED, KEPT)  
  
[Stack]

* + Max Depth = 24
  + Call Chain = \_mutex\_initialize ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#3nqndbk)   osRtxErrorNotify
  + [>>](#1a346fx)   osMutexNew

**\_mutex\_release** (Thumb, 20 bytes, Stack size 8 bytes, rtx\_lib.o(i.\_mutex\_release), UNUSED, KEPT)  
  
[Stack]

* + Max Depth = 24
  + Call Chain = \_mutex\_release ⇒ os\_kernel\_is\_active ⇒ osKernelGetState

[Calls]

* + [>>](#3u2rp3q)   osMutexRelease
  + [>>](#i17xr6)   os\_kernel\_is\_active

**\_platform\_post\_stackheap\_init** (Thumb, 8 bytes, Stack size 8 bytes, rtx\_lib.o(i.\_platform\_post\_stackheap\_init))  
  
[Stack]

* + Max Depth = 16
  + Call Chain = \_platform\_post\_stackheap\_init ⇒ osKernelInitialize

[Calls]

* + [>>](#42ddq1a)   osKernelInitialize

[Called By]

* + [>>](#vx1227)   \_\_rt\_entry\_postsh\_1

**buttonThread** (Thumb, 90 bytes, Stack size 0 bytes, main.o(i.buttonThread))  
  
[Stack]

* + Max Depth = 48
  + Call Chain = buttonThread ⇒ osEventFlagsSet ⇒ osRtxPostProcess ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#243i4a2)   osEventFlagsSet
  + [>>](#14ykbeg)   osDelay
  + [>>](#49gfa85)   isPressed

[Address Reference Count : 1]

* + main.o(i.main)

**configureGPIOinput** (Thumb, 42 bytes, Stack size 0 bytes, gpio.o(i.configureGPIOinput))  
  
[Called By]

* + [>>](#13qzunr)   main

**configureGPIOoutput** (Thumb, 140 bytes, Stack size 0 bytes, gpio.o(i.configureGPIOoutput))  
  
[Called By]

* + [>>](#13qzunr)   main

**greenLEDOnOff** (Thumb, 22 bytes, Stack size 0 bytes, gpio.o(i.greenLEDOnOff))  
  
[Called By]

* + [>>](#2olpkfy)   ledsThread

**isPressed** (Thumb, 20 bytes, Stack size 0 bytes, gpio.o(i.isPressed))  
  
[Called By]

* + [>>](#2b6jogx)   buttonThread

**ledsThread** (Thumb, 224 bytes, Stack size 16 bytes, main.o(i.ledsThread))  
  
[Stack]

* + Max Depth = 48
  + Call Chain = ledsThread ⇒ osEventFlagsWait ⇒ EventFlagsCheck

[Calls]

* + [>>](#22vxnjd)   redLEDOnOff
  + [>>](#1idq7dh)   osKernelGetTickCount
  + [>>](#j8sehv)   osEventFlagsWait
  + [>>](#1pgrrkc)   greenLEDOnOff

[Address Reference Count : 1]

* + main.o(i.main)

**main** (Thumb, 62 bytes, Stack size 0 bytes, main.o(i.main))  
  
[Stack]

* + Max Depth = 24
  + Call Chain = main ⇒ SystemCoreClockUpdate ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#44bvf6o)   osThreadNew
  + [>>](#2hio093)   osKernelStart
  + [>>](#42ddq1a)   osKernelInitialize
  + [>>](#3oy7u29)   osEventFlagsNew
  + [>>](#3abhhcj)   configureGPIOoutput
  + [>>](#qbtyoq)   configureGPIOinput
  + [>>](#2jh5peh)   SystemCoreClockUpdate

[Called By]

* + [>>](#1v1yuxt)   \_\_rt\_entry\_main

**osRtxErrorNotify** (Thumb, 32 bytes, Stack size 0 bytes, rtx\_config.o(i.osRtxErrorNotify))  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#3im3ia3)   \_\_ARM\_common\_switch8

[Called By]

* + [>>](#2wwbldi)   \_mutex\_initialize
  + [>>](#3xzr3ei)   osRtxTimerTick
  + [>>](#4cmhg48)   osRtxThreadStackCheck
  + [>>](#20xfydz)   osRtxPostProcess

**redLEDOnOff** (Thumb, 22 bytes, Stack size 0 bytes, gpio.o(i.redLEDOnOff))  
  
[Called By]

* + [>>](#2olpkfy)   ledsThread

Local Symbols**os\_kernel\_is\_active** (Thumb, 30 bytes, Stack size 8 bytes, rtx\_lib.o(i.os\_kernel\_is\_active), UNUSED, KEPT)  
  
[Stack]

* + Max Depth = 16
  + Call Chain = os\_kernel\_is\_active ⇒ osKernelGetState

[Calls]

* + [>>](#338fx5o)   osKernelGetState

[Called By]

* + [>>](#1c1lvlb)   \_mutex\_release
  + [>>](#1xrdshw)   \_mutex\_acquire

**svcRtxKernelGetState** (Thumb, 12 bytes, Stack size 0 bytes, rtx\_kernel.o(.text.svcRtxKernelGetState))  
[Address Reference Count : 1]

* + rtx\_kernel.o(.text.osKernelGetState)

**svcRtxKernelGetTickCount** (Thumb, 12 bytes, Stack size 0 bytes, rtx\_kernel.o(.text.svcRtxKernelGetTickCount))  
[Address Reference Count : 1]

* + rtx\_kernel.o(.text.osKernelGetTickCount)

**svcRtxKernelInitialize** (Thumb, 380 bytes, Stack size 24 bytes, rtx\_kernel.o(.text.svcRtxKernelInitialize))  
  
[Stack]

* + Max Depth = 40
  + Call Chain = svcRtxKernelInitialize ⇒ osRtxMemoryPoolInit

[Calls]

* + [>>](#2mn7vak)   osRtxMemoryPoolInit
  + [>>](#38czs75)   osRtxMemoryInit
  + [>>](#3ep43zb)   \_\_aeabi\_memclr4

[Address Reference Count : 1]

* + rtx\_kernel.o(.text.osKernelInitialize)

**svcRtxKernelStart** (Thumb, 136 bytes, Stack size 16 bytes, rtx\_kernel.o(.text.svcRtxKernelStart))  
  
[Stack]

* + Max Depth = 120
  + Call Chain = svcRtxKernelStart ⇒ osRtxThreadStartup ⇒ svcRtxThreadNew ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#16x20ju)   osRtxThreadSwitch
  + [>>](#2rrrqc1)   osRtxThreadStartup
  + [>>](#1f7o1he)   osRtxThreadListGet
  + [>>](#2pta16n)   OS\_Tick\_Setup
  + [>>](#4anzqyu)   OS\_Tick\_GetIRQn
  + [>>](#1qoc8b1)   OS\_Tick\_Enable

[Address Reference Count : 1]

* + rtx\_kernel.o(.text.osKernelStart)

**ThreadFlagsCheck** (Thumb, 82 bytes, Stack size 16 bytes, rtx\_thread.o(.text.ThreadFlagsCheck))  
  
[Stack]

* + Max Depth = 16
  + Call Chain = ThreadFlagsCheck

[Called By]

* + [>>](#18vjpp8)   osRtxThreadPostProcess

**osRtxThreadBlock** (Thumb, 60 bytes, Stack size 16 bytes, rtx\_thread.o(.text.osRtxThreadBlock))  
  
[Stack]

* + Max Depth = 16
  + Call Chain = osRtxThreadBlock

[Called By]

* + [>>](#302dr9l)   osRtxThreadDispatch

**osRtxThreadDelayInsert** (Thumb, 140 bytes, Stack size 16 bytes, rtx\_thread.o(.text.osRtxThreadDelayInsert))  
  
[Stack]

* + Max Depth = 16
  + Call Chain = osRtxThreadDelayInsert

[Called By]

* + [>>](#3qwpj7n)   osRtxThreadWaitEnter

**osRtxThreadDelayRemove** (Thumb, 96 bytes, Stack size 8 bytes, rtx\_thread.o(.text.osRtxThreadDelayRemove))  
  
[Stack]

* + Max Depth = 8
  + Call Chain = osRtxThreadDelayRemove

[Called By]

* + [>>](#261ztfg)   osRtxThreadWaitExit

**osRtxThreadFree** (Thumb, 84 bytes, Stack size 8 bytes, rtx\_thread.o(.text.osRtxThreadFree))  
  
[Stack]

* + Max Depth = 28
  + Call Chain = osRtxThreadFree ⇒ osRtxMemoryFree

[Calls]

* + [>>](#47hxl2r)   osRtxMemoryPoolFree
  + [>>](#odc9jc)   osRtxMemoryFree

[Called By]

* + [>>](#3sv78d1)   svcRtxThreadExit

**osRtxThreadPostProcess** (Thumb, 38 bytes, Stack size 8 bytes, rtx\_thread.o(.text.osRtxThreadPostProcess))  
  
[Stack]

* + Max Depth = 64
  + Call Chain = osRtxThreadPostProcess ⇒ osRtxThreadWaitExit ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#261ztfg)   osRtxThreadWaitExit
  + [>>](#vgdtq7)   ThreadFlagsCheck

[Address Reference Count : 1]

* + rtx\_thread.o(.text.svcRtxThreadNew)

**svcRtxThreadExit** (Thumb, 108 bytes, Stack size 16 bytes, rtx\_thread.o(.text.svcRtxThreadExit))  
  
[Stack]

* + Max Depth = 96
  + Call Chain = svcRtxThreadExit ⇒ osRtxMutexOwnerRelease ⇒ osRtxThreadWaitExit ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#16x20ju)   osRtxThreadSwitch
  + [>>](#1f7o1he)   osRtxThreadListGet
  + [>>](#261ztfg)   osRtxThreadWaitExit
  + [>>](#11si5id)   osRtxMutexOwnerRelease
  + [>>](#2tq9fhf)   osRtxThreadFree

[Address Reference Count : 1]

* + rtx\_thread.o(.text.osThreadExit)

**svcRtxThreadNew** (Thumb, 512 bytes, Stack size 48 bytes, rtx\_thread.o(.text.svcRtxThreadNew))  
  
[Stack]

* + Max Depth = 88
  + Call Chain = svcRtxThreadNew ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#302dr9l)   osRtxThreadDispatch
  + [>>](#47hxl2r)   osRtxMemoryPoolFree
  + [>>](#1nia2ey)   osRtxMemoryPoolAlloc
  + [>>](#odc9jc)   osRtxMemoryFree
  + [>>](#2981zbj)   osRtxMemoryAlloc
  + [>>](#3ep43zb)   \_\_aeabi\_memclr4
  + [>>](#2fk6b3p)   \_\_aeabi\_memclr

[Called By]

* + [>>](#2rrrqc1)   osRtxThreadStartup

[Address Reference Count : 1]

* + rtx\_thread.o(.text.osThreadNew)

**svcRtxDelay** (Thumb, 18 bytes, Stack size 8 bytes, rtx\_delay.o(.text.svcRtxDelay))  
  
[Stack]

* + Max Depth = 48
  + Call Chain = svcRtxDelay ⇒ osRtxThreadWaitEnter ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#3qwpj7n)   osRtxThreadWaitEnter

[Address Reference Count : 1]

* + rtx\_delay.o(.text.osDelay)

**EventFlagsCheck** (Thumb, 82 bytes, Stack size 16 bytes, rtx\_evflags.o(.text.EventFlagsCheck))  
  
[Stack]

* + Max Depth = 16
  + Call Chain = EventFlagsCheck

[Called By]

* + [>>](#j8sehv)   osEventFlagsWait
  + [>>](#10kxoro)   svcRtxEventFlagsWait
  + [>>](#2lfnejv)   svcRtxEventFlagsSet
  + [>>](#1maplo9)   osRtxEventFlagsPostProcess

**osRtxEventFlagsPostProcess** (Thumb, 58 bytes, Stack size 24 bytes, rtx\_evflags.o(.text.osRtxEventFlagsPostProcess))  
  
[Stack]

* + Max Depth = 80
  + Call Chain = osRtxEventFlagsPostProcess ⇒ osRtxThreadWaitExit ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#375fbgg)   EventFlagsCheck
  + [>>](#261ztfg)   osRtxThreadWaitExit
  + [>>](#2eclud0)   osRtxThreadListRemove

[Address Reference Count : 1]

* + rtx\_evflags.o(.text.svcRtxEventFlagsNew)

**svcRtxEventFlagsNew** (Thumb, 116 bytes, Stack size 8 bytes, rtx\_evflags.o(.text.svcRtxEventFlagsNew))  
  
[Stack]

* + Max Depth = 36
  + Call Chain = svcRtxEventFlagsNew ⇒ osRtxMemoryAlloc

[Calls]

* + [>>](#1nia2ey)   osRtxMemoryPoolAlloc
  + [>>](#2981zbj)   osRtxMemoryAlloc

[Address Reference Count : 1]

* + rtx\_evflags.o(.text.osEventFlagsNew)

**svcRtxEventFlagsSet** (Thumb, 144 bytes, Stack size 32 bytes, rtx\_evflags.o(.text.svcRtxEventFlagsSet))  
  
[Stack]

* + Max Depth = 88
  + Call Chain = svcRtxEventFlagsSet ⇒ osRtxThreadWaitExit ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#302dr9l)   osRtxThreadDispatch
  + [>>](#375fbgg)   EventFlagsCheck
  + [>>](#261ztfg)   osRtxThreadWaitExit
  + [>>](#2eclud0)   osRtxThreadListRemove

[Address Reference Count : 1]

* + rtx\_evflags.o(.text.osEventFlagsSet)

**svcRtxEventFlagsWait** (Thumb, 120 bytes, Stack size 32 bytes, rtx\_evflags.o(.text.svcRtxEventFlagsWait))  
  
[Stack]

* + Max Depth = 72
  + Call Chain = svcRtxEventFlagsWait ⇒ osRtxThreadWaitEnter ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#375fbgg)   EventFlagsCheck
  + [>>](#3qwpj7n)   osRtxThreadWaitEnter
  + [>>](#3z7bk57)   osRtxThreadListPut

[Address Reference Count : 1]

* + rtx\_evflags.o(.text.osEventFlagsWait)

**svcRtxMutexAcquire** (Thumb, 160 bytes, Stack size 24 bytes, rtx\_mutex.o(.text.svcRtxMutexAcquire))  
  
[Stack]

* + Max Depth = 64
  + Call Chain = svcRtxMutexAcquire ⇒ osRtxThreadWaitEnter ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#3qwpj7n)   osRtxThreadWaitEnter
  + [>>](#thw4kt)   osRtxThreadListSort
  + [>>](#3z7bk57)   osRtxThreadListPut

[Address Reference Count : 1]

* + rtx\_mutex.o(.text.osMutexAcquire)

**svcRtxMutexDelete** (Thumb, 208 bytes, Stack size 24 bytes, rtx\_mutex.o(.text.svcRtxMutexDelete))  
  
[Stack]

* + Max Depth = 80
  + Call Chain = svcRtxMutexDelete ⇒ osRtxThreadWaitExit ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#1f7o1he)   osRtxThreadListGet
  + [>>](#302dr9l)   osRtxThreadDispatch
  + [>>](#261ztfg)   osRtxThreadWaitExit
  + [>>](#thw4kt)   osRtxThreadListSort
  + [>>](#47hxl2r)   osRtxMemoryPoolFree
  + [>>](#odc9jc)   osRtxMemoryFree

[Address Reference Count : 1]

* + rtx\_mutex.o(.text.osMutexDelete)

**svcRtxMutexNew** (Thumb, 120 bytes, Stack size 16 bytes, rtx\_mutex.o(.text.svcRtxMutexNew))  
  
[Stack]

* + Max Depth = 44
  + Call Chain = svcRtxMutexNew ⇒ osRtxMemoryAlloc

[Calls]

* + [>>](#1nia2ey)   osRtxMemoryPoolAlloc
  + [>>](#2981zbj)   osRtxMemoryAlloc
  + [>>](#3ep43zb)   \_\_aeabi\_memclr4

[Address Reference Count : 1]

* + rtx\_mutex.o(.text.osMutexNew)

**svcRtxMutexRelease** (Thumb, 192 bytes, Stack size 16 bytes, rtx\_mutex.o(.text.svcRtxMutexRelease))  
  
[Stack]

* + Max Depth = 72
  + Call Chain = svcRtxMutexRelease ⇒ osRtxThreadWaitExit ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#1f7o1he)   osRtxThreadListGet
  + [>>](#302dr9l)   osRtxThreadDispatch
  + [>>](#261ztfg)   osRtxThreadWaitExit

[Address Reference Count : 1]

* + rtx\_mutex.o(.text.osMutexRelease)

**TimerInsert** (Thumb, 76 bytes, Stack size 16 bytes, rtx\_timer.o(.text.TimerInsert))  
  
[Stack]

* + Max Depth = 16
  + Call Chain = TimerInsert

[Called By]

* + [>>](#3xzr3ei)   osRtxTimerTick

**osRtxTimerTick** (Thumb, 96 bytes, Stack size 16 bytes, rtx\_timer.o(.text.osRtxTimerTick))  
  
[Stack]

* + Max Depth = 72
  + Call Chain = osRtxTimerTick ⇒ osMessageQueuePut ⇒ osRtxPostProcess ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#3nqndbk)   osRtxErrorNotify
  + [>>](#1vsw3ci)   osMessageQueuePut
  + [>>](#1e03kqp)   TimerInsert

[Address Reference Count : 1]

* + rtx\_timer.o(.text.osRtxTimerThread)

**MessageQueueGet** (Thumb, 76 bytes, Stack size 8 bytes, rtx\_msgqueue.o(.text.MessageQueueGet))  
  
[Stack]

* + Max Depth = 8
  + Call Chain = MessageQueueGet

[Called By]

* + [>>](#wnyagw)   osMessageQueueGet
  + [>>](#4bewzdj)   svcRtxMessageQueueGet

**MessageQueuePut** (Thumb, 96 bytes, Stack size 20 bytes, rtx\_msgqueue.o(.text.MessageQueuePut))  
  
[Stack]

* + Max Depth = 20
  + Call Chain = MessageQueuePut

[Called By]

* + [>>](#15phjt5)   svcRtxMessageQueuePut
  + [>>](#4bewzdj)   svcRtxMessageQueueGet
  + [>>](#1rf9gpq)   osRtxMessageQueuePostProcess

**MessageQueueRemove** (Thumb, 40 bytes, Stack size 8 bytes, rtx\_msgqueue.o(.text.MessageQueueRemove))  
  
[Stack]

* + Max Depth = 8
  + Call Chain = MessageQueueRemove

[Called By]

* + [>>](#4bewzdj)   svcRtxMessageQueueGet
  + [>>](#1rf9gpq)   osRtxMessageQueuePostProcess

**osRtxMessageQueuePostProcess** (Thumb, 222 bytes, Stack size 24 bytes, rtx\_msgqueue.o(.text.osRtxMessageQueuePostProcess))  
  
[Stack]

* + Max Depth = 80
  + Call Chain = osRtxMessageQueuePostProcess ⇒ osRtxThreadWaitExit ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#1f7o1he)   osRtxThreadListGet
  + [>>](#261ztfg)   osRtxThreadWaitExit
  + [>>](#1smtxgf)   osRtxThreadRegPtr
  + [>>](#47hxl2r)   osRtxMemoryPoolFree
  + [>>](#1nia2ey)   osRtxMemoryPoolAlloc
  + [>>](#haapch)   \_\_aeabi\_memcpy
  + [>>](#3c9z6hx)   MessageQueueRemove
  + [>>](#sabnu4)   MessageQueuePut

[Address Reference Count : 1]

* + rtx\_msgqueue.o(.text.svcRtxMessageQueueNew)

**svcRtxMessageQueueGet** (Thumb, 240 bytes, Stack size 32 bytes, rtx\_msgqueue.o(.text.svcRtxMessageQueueGet))  
  
[Stack]

* + Max Depth = 88
  + Call Chain = svcRtxMessageQueueGet ⇒ osRtxThreadWaitExit ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#1f7o1he)   osRtxThreadListGet
  + [>>](#261ztfg)   osRtxThreadWaitExit
  + [>>](#3qwpj7n)   osRtxThreadWaitEnter
  + [>>](#1smtxgf)   osRtxThreadRegPtr
  + [>>](#3z7bk57)   osRtxThreadListPut
  + [>>](#47hxl2r)   osRtxMemoryPoolFree
  + [>>](#1nia2ey)   osRtxMemoryPoolAlloc
  + [>>](#haapch)   \_\_aeabi\_memcpy
  + [>>](#3c9z6hx)   MessageQueueRemove
  + [>>](#sabnu4)   MessageQueuePut
  + [>>](#2d51dmb)   MessageQueueGet

[Address Reference Count : 1]

* + rtx\_msgqueue.o(.text.osMessageQueueGet)

**svcRtxMessageQueueNew** (Thumb, 468 bytes, Stack size 48 bytes, rtx\_msgqueue.o(.text.svcRtxMessageQueueNew))  
  
[Stack]

* + Max Depth = 76
  + Call Chain = svcRtxMessageQueueNew ⇒ osRtxMemoryAlloc

[Calls]

* + [>>](#2mn7vak)   osRtxMemoryPoolInit
  + [>>](#47hxl2r)   osRtxMemoryPoolFree
  + [>>](#1nia2ey)   osRtxMemoryPoolAlloc
  + [>>](#odc9jc)   osRtxMemoryFree
  + [>>](#2981zbj)   osRtxMemoryAlloc
  + [>>](#2fk6b3p)   \_\_aeabi\_memclr

[Address Reference Count : 1]

* + rtx\_msgqueue.o(.text.osMessageQueueNew)

**svcRtxMessageQueuePut** (Thumb, 208 bytes, Stack size 32 bytes, rtx\_msgqueue.o(.text.svcRtxMessageQueuePut))  
  
[Stack]

* + Max Depth = 88
  + Call Chain = svcRtxMessageQueuePut ⇒ osRtxThreadWaitExit ⇒ osRtxThreadDispatch ⇒ osRtxThreadSwitch ⇒ osRtxThreadStackCheck ⇒ osRtxErrorNotify ⇒ \_\_ARM\_common\_switch8

[Calls]

* + [>>](#1f7o1he)   osRtxThreadListGet
  + [>>](#261ztfg)   osRtxThreadWaitExit
  + [>>](#3qwpj7n)   osRtxThreadWaitEnter
  + [>>](#1smtxgf)   osRtxThreadRegPtr
  + [>>](#3z7bk57)   osRtxThreadListPut
  + [>>](#1nia2ey)   osRtxMemoryPoolAlloc
  + [>>](#haapch)   \_\_aeabi\_memcpy
  + [>>](#sabnu4)   MessageQueuePut

[Address Reference Count : 1]

* + rtx\_msgqueue.o(.text.osMessageQueuePut)

Undefined Global Symbols**\_call\_atexit\_fns** (ARM, 0 bytes, Stack size 0 bytes, UNDEFINED)  
  
[Called By]

* + [>>](#meukdy)   exit