```
  /* Get size of rtc slow data based on rtc_data_location alias */
  _rtc_slow_length = (ORIGIN(rtc_slow_seg) == ORIGIN(rtc_data_location))
                        ? (_rtc_force_slow_end - _rtc_data_start)
                        : (_rtc_force_slow_end - _rtc_force_slow_start);

  _rtc_fast_length = (ORIGIN(rtc_slow_seg) == ORIGIN(rtc_data_location))
                        ? (_rtc_force_fast_end - _rtc_fast_start)
                        : (_rtc_noinit_end - _rtc_fast_start);

  ASSERT((_rtc_slow_length <= LENGTH(rtc_slow_seg)),
          "RTC_SLOW segment data does not fit.")

  ASSERT((_rtc_fast_length <= LENGTH(rtc_data_seg)),
          "RTC_FAST segment data does not fit.")

  /* Send .iram0 code to iram */
  .iram0.vectors :
  {
    _iram_start = ABSOLUTE(.);
    /* Vectors go to IRAM */
    _vector_table = ABSOLUTE(.);
    /* Vectors according to builds/RF-2015.2-win32/esp108_v1_2_s5_512int_2/config.html */
    . = 0x0;
    KEEP(*(.WindowVectors.text));
    . = 0x180;
    KEEP(*(.Level2InterruptVector.text));
    . = 0x1c0;
    KEEP(*(.Level3InterruptVector.text));
    . = 0x200;
    KEEP(*(.Level4InterruptVector.text));
    . = 0x240;
    KEEP(*(.Level5InterruptVector.text));
    . = 0x280;
    KEEP(*(.DebugExceptionVector.text));
    . = 0x2c0;
    KEEP(*(.NMIExceptionVector.text));
    . = 0x300;
    KEEP(*(.KernelExceptionVector.text));
    . = 0x340;
    KEEP(*(.UserExceptionVector.text));
    . = 0x3C0;
    KEEP(*(.DoubleExceptionVector.text));
    . = 0x400;
    _invalid_pc_placeholder = ABSOLUTE(.);
    *(.*Vector.literal)

    *(.UserEnter.literal);
    *(.UserEnter.text);
    . = ALIGN (16);
    *(.entry.text)
    *(.init.literal)
    *(.init)

    _init_end = ABSOLUTE(.);
  } > iram0_0_seg

  .iram0.text :
  {
    /* Code marked as runnning out of IRAM */
    _iram_text_start = ABSOLUTE(.);

    *(.iram1 .iram1.*)
    *libapp_trace.a:app_trace.*(.literal .literal.* .text .text.*)
    *libapp_trace.a:app_trace_util.*(.literal .literal.* .text .text.*)
    *libesp_event.a:default_event_loop.*(.literal.esp_event_isr_post .text.esp_event_isr_post)
    *libesp_event.a:esp_event.*(.literal.esp_event_isr_post_to .text.esp_event_isr_post_to)
    *libesp_hw_support.a:cpu_util.*(.literal .literal.* .text .text.*)
```

```
  _rtc_slow_reserved_length = _rtc_slow_reserved_end - _rtc_slow_reserved_start;
  _rtc_reserved_length = _rtc_slow_reserved_length;
  ASSERT((_rtc_slow_reserved_length <= LENGTH(rtc_slow_reserved_seg)),
          "RTC SLOW reserved segment data does not fit.")

  /* Get size of rtc slow data based on rtc_data_location alias */
  _rtc_slow_length = (ORIGIN(rtc_slow_seg) == ORIGIN(rtc_data_location))
                        ? (_rtc_force_slow_end - _rtc_data_start)
                        : (_rtc_force_slow_end - _rtc_force_slow_start);

  _rtc_fast_length = (ORIGIN(rtc_slow_seg) == ORIGIN(rtc_data_location))
                        ? (_rtc_force_fast_end - _rtc_fast_start)
                        : (_rtc_noinit_end - _rtc_fast_start);

  ASSERT((_rtc_slow_length <= LENGTH(rtc_slow_seg)),
          "RTC_SLOW segment data does not fit.")

  ASSERT((_rtc_fast_length <= LENGTH(rtc_data_seg)),
          "RTC_FAST segment data does not fit.")

  /* Send .iram0 code to iram */
  .iram0.vectors :
  {
    _iram_start = ABSOLUTE(.);
    /* Vectors go to IRAM */
    _vector_table = ABSOLUTE(.);
    /* Vectors according to builds/RF-2015.2-win32/esp108_v1_2_s5_512int_2/config.html */
    . = 0x0;
    KEEP(*(.WindowVectors.text));
    . = 0x180;
    KEEP(*(.Level2InterruptVector.text));
    . = 0x1c0;
    KEEP(*(.Level3InterruptVector.text));
    . = 0x200;
    KEEP(*(.Level4InterruptVector.text));
    . = 0x240;
    KEEP(*(.Level5InterruptVector.text));
    . = 0x280;
    KEEP(*(.DebugExceptionVector.text));
    . = 0x2c0;
    KEEP(*(.NMIExceptionVector.text));
    . = 0x300;
    KEEP(*(.KernelExceptionVector.text));
    . = 0x340;
    KEEP(*(.UserExceptionVector.text));
    . = 0x3C0;
    KEEP(*(.DoubleExceptionVector.text));
    . = 0x400;
    _invalid_pc_placeholder = ABSOLUTE(.);
    *(.*Vector.literal)

    *(.UserEnter.literal);
    *(.UserEnter.text);
    . = ALIGN (16);
    *(.entry.text)
    *(.init.literal)
    *(.init)

    _init_end = ABSOLUTE(.);
  } > iram0_0_seg

  .iram0.text :
  {
    /* Code marked as runnning out of IRAM */
    _iram_text_start = ABSOLUTE(.);

    *(.iram1 .iram1.*)
    *libapp_trace.a:app_trace.*(.literal .literal.* .text .text.*)
    *libapp_trace.a:app_trace_util.*(.literal .literal.* .text .text.*)
    *libapp_trace.a:port_uart.*(.literal .literal.* .text .text.*)
    *libesp_event.a:default_event_loop.*(.literal.esp_event_isr_post .text.esp_event_isr_post)
    *libesp_event.a:esp_event.*(.literal.esp_event_isr_post_to .text.esp_event_isr_post_to)
    *libesp_hw_support.a:cpu.*(.literal.esp_cpu_compare_and_set .text.esp_cpu_compare_and_set)
    *libesp_hw_support.a:cpu.*(.literal.esp_cpu_reset .text.esp_cpu_reset)
    *libesp_hw_support.a:cpu.*(.literal.esp_cpu_stall .text.esp_cpu_stall)
    *libesp_hw_support.a:cpu.*(.literal.esp_cpu_unstall .text.esp_cpu_unstall)
    *libesp_hw_support.a:cpu.*(.literal.esp_cpu_wait_for_intr .text.esp_cpu_wait_for_intr)
    *libesp_hw_support.a:esp_gpio_reserve.*(.literal.esp_gpio_is_pin_reserved .text.esp_gpio_is_pin_reser
```

```
                                                                                *libesp_hw_support.a:esp_gpio_reserve.*(.literal.esp_gpio_reserve_pins .text.esp_gpio_reserve_pins)
                                                                                *libesp_hw_support.a:esp_memory_utils.*(.literal .literal.* .text .text.*)
                                                                                *libesp_hw_support.a:periph_ctrl.*(.literal.periph_module_reset .text.periph_module_reset)
                                                                                *libesp_hw_support.a:periph_ctrl.*(.literal.wifi_module_disable .text.wifi_module_disable)
                                                                                *libesp_hw_support.a:periph_ctrl.*(.literal.wifi_module_enable .text.wifi_module_enable)
    *libesp_hw_support.a:rtc_clk.*(.literal .literal.* .text .text.*)            *libesp_hw_support.a:rtc_clk.*(.literal .literal.* .text .text.*)
    *libesp_hw_support.a:rtc_init.*(.literal.rtc_vddsdio_set_config .text.rtc_vddsdio_set_config)
    *libesp_hw_support.a:rtc_pm.*(.literal .literal.* .text .text.*)
    *libesp_hw_support.a:rtc_sleep.*(.literal .literal.* .text .text.*)          *libesp_hw_support.a:rtc_sleep.*(.literal .literal.* .text .text.*)
    *libesp_hw_support.a:rtc_time.*(.literal .literal.* .text .text.*)           *libesp_hw_support.a:rtc_time.*(.literal .literal.* .text .text.*)
    *libesp_hw_support.a:rtc_wdt.*(.literal .literal.* .text .text.*)            *libesp_hw_support.a:rtc_wdt.*(.literal .literal.* .text .text.*)
                                                                                *libesp_hw_support.a:sar_periph_ctrl.*(.literal.sar_periph_ctrl_power_enable .text.sar_periph_ctrl_po
                                                                                *libesp_mm.a:cache_esp32.*(.literal .literal.* .text .text.*)
                                                                                *libesp_mm.a:esp_cache.*(.literal .literal.* .text .text.*)
    *libesp_ringbuf.a:(.literal .literal.* .text .text.*)                       *libesp_ringbuf.a:(.literal .literal.* .text .text.*)
                                                                                *libesp_rom.a:esp_rom_spiflash.*(.literal .literal.* .text .text.*)
    *libesp_system.a:esp_err.*(.literal .literal.* .text .text.*)               *libesp_system.a:esp_err.*(.literal .literal.* .text .text.*)
    *libesp_system.a:esp_system.*(.literal.esp_system_abort .text.esp_system_abort)  *libesp_system.a:esp_system_chip.*(.literal.esp_system_abort .text.esp_system_abort)
    *libesp_system.a:ubsan.*(.literal .literal.* .text .text.*)                 *libesp_system.a:ubsan.*(.literal .literal.* .text .text.*)
    *libfreertos.a:(EXCLUDE_FILE(*libfreertos.a:port.* *libfreertos.a:port_common.*) .literal EXCLUDE_FI  *libesp_wifi.a:esp_adapter.*(.literal.coex_pti_get_wrapper .text.coex_pti_get_wrapper)
    *libfreertos.a:port.*(.literal.pxPortInitialiseStack .literal.unlikely.vPortEndScheduler .literal.vA  *libesp_wifi.a:wifi_netif.*(.literal.wifi_sta_receive .text.wifi_sta_receive)
    *libfreertos.a:port_common.*(.literal.esp_startup_start_app_common .literal.other_cpu_startup_idle_h  *libesp_wifi.a:wifi_netif.*(.literal.wifi_transmit_wrap .text.wifi_transmit_wrap)
                                                                                *libfreertos.a:(.literal .literal.* .text .text.*)
    *libgcc.a:lib2funcs.*(.literal .literal.* .text .text.*)                    *libgcc.a:lib2funcs.*(.literal .literal.* .text .text.*)
    *libgcov.a:(.literal .literal.* .text .text.*)                              *libgcov.a:(.literal .literal.* .text .text.*)
    *libhal.a:cpu_hal.*(.literal .literal.* .text .text.*)                      *libhal.a:cache_hal_esp32.*(.literal .literal.* .text .text.*)
    *libhal.a:i2c_hal_iram.*(.literal .literal.* .text .text.*)                 *libhal.a:i2c_hal_iram.*(.literal .literal.* .text .text.*)
    *libhal.a:ledc_hal_iram.*(.literal .literal.* .text .text.*)                *libhal.a:ledc_hal_iram.*(.literal .literal.* .text .text.*)
    *libhal.a:soc_hal.*(.literal .literal.* .text .text.*)                      *libhal.a:mmu_hal.*(.literal .literal.* .text .text.*)
    *libhal.a:spi_flash_encrypt_hal_iram.*(.literal .literal.* .text .text.*)   *libhal.a:spi_flash_encrypt_hal_iram.*(.literal .literal.* .text .text.*)
    *libhal.a:spi_flash_hal_iram.*(.literal .literal.* .text .text.*)           *libhal.a:spi_flash_hal_iram.*(.literal .literal.* .text .text.*)
    *libhal.a:spi_hal_iram.*(.literal .literal.* .text .text.*)                 *libhal.a:spi_hal_iram.*(.literal .literal.* .text .text.*)
    *libhal.a:spi_slave_hal_iram.*(.literal .literal.* .text .text.*)           *libhal.a:spi_slave_hal_iram.*(.literal .literal.* .text .text.*)
    *libhal.a:wdt_hal_iram.*(.literal .literal.* .text .text.*)                 *libhal.a:wdt_hal_iram.*(.literal .literal.* .text .text.*)
    *libheap.a:heap_tlsf.*(.literal .literal.* .text .text.*)                   *libheap.a:multi_heap.*(.literal.assert_valid_block .text.assert_valid_block)
    *libheap.a:multi_heap.*(.literal .literal.* .text .text.*)                  *libheap.a:multi_heap.*(.literal.multi_heap_aligned_alloc_impl .text.multi_heap_aligned_alloc_impl)
                                                                                *libheap.a:multi_heap.*(.literal.multi_heap_aligned_alloc_impl_offs .text.multi_heap_aligned_alloc_im
                                                                                *libheap.a:multi_heap.*(.literal.multi_heap_free_impl .text.multi_heap_free_impl)
                                                                                *libheap.a:multi_heap.*(.literal.multi_heap_get_allocated_size_impl .text.multi_heap_get_allocated_si
                                                                                *libheap.a:multi_heap.*(.literal.multi_heap_get_block_address_impl .text.multi_heap_get_block_address
                                                                                *libheap.a:multi_heap.*(.literal.multi_heap_get_first_block .text.multi_heap_get_first_block)
                                                                                *libheap.a:multi_heap.*(.literal.multi_heap_get_next_block .text.multi_heap_get_next_block)
                                                                                *libheap.a:multi_heap.*(.literal.multi_heap_internal_lock .text.multi_heap_internal_lock)
                                                                                *libheap.a:multi_heap.*(.literal.multi_heap_internal_unlock .text.multi_heap_internal_unlock)
                                                                                *libheap.a:multi_heap.*(.literal.multi_heap_is_free .text.multi_heap_is_free)
                                                                                *libheap.a:multi_heap.*(.literal.multi_heap_malloc_impl .text.multi_heap_malloc_impl)
                                                                                *libheap.a:multi_heap.*(.literal.multi_heap_realloc_impl .text.multi_heap_realloc_impl)
                                                                                *libheap.a:multi_heap.*(.literal.multi_heap_set_lock .text.multi_heap_set_lock)
                                                                                *libheap.a:tlsf.*(.literal.tlsf_align_size .text.tlsf_align_size)
                                                                                *libheap.a:tlsf.*(.literal.tlsf_alloc_overhead .text.tlsf_alloc_overhead)
                                                                                *libheap.a:tlsf.*(.literal.tlsf_block_size .text.tlsf_block_size)
                                                                                *libheap.a:tlsf.*(.literal.tlsf_block_size_max .text.tlsf_block_size_max)
                                                                                *libheap.a:tlsf.*(.literal.tlsf_block_size_min .text.tlsf_block_size_min)
                                                                                *libheap.a:tlsf.*(.literal.tlsf_free .text.tlsf_free)
                                                                                *libheap.a:tlsf.*(.literal.tlsf_get_pool .text.tlsf_get_pool)
                                                                                *libheap.a:tlsf.*(.literal.tlsf_malloc .text.tlsf_malloc)
                                                                                *libheap.a:tlsf.*(.literal.tlsf_memalign .text.tlsf_memalign)
                                                                                *libheap.a:tlsf.*(.literal.tlsf_memalign_offs .text.tlsf_memalign_offs)
                                                                                *libheap.a:tlsf.*(.literal.tlsf_realloc .text.tlsf_realloc)
                                                                                *libheap.a:tlsf.*(.literal.tlsf_size .text.tlsf_size)
    *liblog.a:log.*(.literal.esp_log_write .text.esp_log_write)                 *liblog.a:log.*(.literal.esp_log_write .text.esp_log_write)
    *liblog.a:log_freertos.*(.literal.esp_log_early_timestamp .text.esp_log_early_timestamp)  *liblog.a:log_freertos.*(.literal.esp_log_early_timestamp .text.esp_log_early_timestamp)
    *liblog.a:log_freertos.*(.literal.esp_log_impl_lock .text.esp_log_impl_lock)  *liblog.a:log_freertos.*(.literal.esp_log_impl_lock .text.esp_log_impl_lock)
    *liblog.a:log_freertos.*(.literal.esp_log_impl_lock_timeout .text.esp_log_impl_lock_timeout)  *liblog.a:log_freertos.*(.literal.esp_log_impl_lock_timeout .text.esp_log_impl_lock_timeout)
    *liblog.a:log_freertos.*(.literal.esp_log_impl_unlock .text.esp_log_impl_unlock)  *liblog.a:log_freertos.*(.literal.esp_log_impl_unlock .text.esp_log_impl_unlock)
    *liblog.a:log_freertos.*(.literal.esp_log_timestamp .text.esp_log_timestamp)  *liblog.a:log_freertos.*(.literal.esp_log_timestamp .text.esp_log_timestamp)
    *libnet80211.a:(.wifi0iram .wifi0iram.*)                                    *libnet80211.a:(.wifi0iram .wifi0iram.*)
    *libnet80211.a:(.wifirxiram .wifirxiram.*)                                  *libnet80211.a:(.wifirxiram .wifirxiram.*)
    *libnet80211.a:(.wifislprxiram .wifislprxiram.*)                            *libnet80211.a:(.wifislprxiram .wifislprxiram.*)
    *libnewlib.a:abort.*(.literal .literal.* .text .text.*)                     *libnewlib.a:abort.*(.literal .literal.* .text .text.*)
    *libnewlib.a:assert.*(.literal .literal.* .text .text.*)                    *libnewlib.a:assert.*(.literal .literal.* .text .text.*)
    *libnewlib.a:heap.*(.literal .literal.* .text .text.*)                      *libnewlib.a:heap.*(.literal .literal.* .text .text.*)
    *libnewlib.a:stdatomic.*(.literal .literal.* .text .text.*)                 *libnewlib.a:stdatomic.*(.literal .literal.* .text .text.*)
    *libpp.a:(.wifi0iram .wifi0iram.*)                                          *libpp.a:(.wifi0iram .wifi0iram.*)
    *libpp.a:(.wifiorslpiram .wifiorslpiram.*)                                  *libpp.a:(.wifiorslpiram .wifiorslpiram.*)
    *libpp.a:(.wifirxiram .wifirxiram.*)                                        *libpp.a:(.wifirxiram .wifirxiram.*)
    *libpp.a:(.wifislprxiram .wifislprxiram.*)                                  *libpp.a:(.wifislprxiram .wifislprxiram.*)
    *librtc.a:(.literal .literal.* .text .text.*)                               *librtc.a:(.literal .literal.* .text .text.*)
```

```
    *libsoc.a:lldesc.*(.literal .literal.* .text .text.*)
                                                                    *libsoc.a:lldesc.*(.literal .literal.* .text .text.*)
                                                                    *libspi_flash.a:flash_brownout_hook.*(.literal .literal.* .text .text.*)
    *libspi_flash.a:memspi_host_driver.*(.literal .literal.* .text .text.*)
    *libspi_flash.a:spi_flash_chip_boya.*(.literal .literal.* .text .text.*)
    *libspi_flash.a:spi_flash_chip_gd.*(.literal .literal.* .text .text.*)
    *libspi_flash.a:spi_flash_chip_generic.*(.literal .literal.* .text .text.*)
    *libspi_flash.a:spi_flash_chip_issi.*(.literal .literal.* .text .text.*)
    *libspi_flash.a:spi_flash_chip_mxic.*(.literal .literal.* .text .text.*)
    *libspi_flash.a:spi_flash_chip_th.*(.literal .literal.* .text .text.*)
    *libspi_flash.a:spi_flash_chip_winbond.*(.literal .literal.* .text .text.*)
    *libspi_flash.a:spi_flash_rom_patch.*(.literal .literal.* .text .text.*)
                                                                    *libspi_flash.a:spi_flash_wrap.*(.literal .literal.* .text .text.*)
    *libxt_hal.a:(.literal .literal.* .text .text.*)
    *libxtensa.a:eri.*(.literal .literal.* .text .text.*)
    *libxtensa.a:xtensa_intr_asm.*(.literal .literal.* .text .text.*)

  } > iram0_0_seg

  .dram0.data :
  {
    _data_start = ABSOLUTE(.);
    *(.gnu.linkonce.d.*)
    *(.data1)
    *(.sdata)
    *(.sdata.*)
    *(.gnu.linkonce.s.*)
    *(.gnu.linkonce.s2.*)
    *(.jcr)

    _esp_system_init_fn_array_start = ABSOLUTE(.);
    KEEP (*(SORT(.esp_system_init_fn) SORT(.esp_system_init_fn.*)))
    _esp_system_init_fn_array_end = ABSOLUTE(.);

    *(EXCLUDE_FILE(*libbt.a *libbtdm_app.a *libnimble.a) .data EXCLUDE_FILE(*libbt.a *libbtdm_app.a *lib
    *(.dram1 .dram1.*)
    _coredump_dram_start = ABSOLUTE(.);
    *(.dram2.coredump .dram2.coredump.*)
    _coredump_dram_end = ABSOLUTE(.);
    *libapp_trace.a:app_trace.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libapp_trace.a:app_trace_util.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *(EXCLUDE_FILE(*libble_app.a *libbt.a *libbtdm_app.a *libnimble.a) .data EXCLUDE_FILE(*libble_app.a *
                                                                    *(.dram1 .dram1.*)
                                                                    _coredump_dram_start = ABSOLUTE(.);
                                                                    *(.dram2.coredump .dram2.coredump.*)
                                                                    _coredump_dram_end = ABSOLUTE(.);
                                                                    *libapp_trace.a:app_trace.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libapp_trace.a:app_trace_util.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libapp_trace.a:port_uart.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    . = ALIGN(4);
                                                                    _bt_controller_data_start = ABSOLUTE(.);
                                                                    *libble_app.a:(.data .data.*)
                                                                    . = ALIGN(4);
                                                                    _bt_controller_data_end = ABSOLUTE(.);
                                                                    . = ALIGN(4);
    _bt_data_start = ABSOLUTE(.);
    *libbt.a:(.data .data.*)
    . = ALIGN(4);
    _bt_data_end = ABSOLUTE(.);
                                                                    . = ALIGN(4);
    _btdm_data_start = ABSOLUTE(.);
    *libbtdm_app.a:(.data .data.*)
    . = ALIGN(4);
    _btdm_data_end = ABSOLUTE(.);
                                                                    *libesp_hw_support.a:esp_memory_utils.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libesp_hw_support.a:rtc_clk.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libesp_mm.a:cache_esp32.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libesp_mm.a:esp_cache.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libesp_rom.a:esp_rom_spiflash.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libesp_system.a:esp_err.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libesp_system.a:ubsan.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libgcov.a:(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libhal.a:cpu_hal.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libhal.a:i2c_hal_iram.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libhal.a:ledc_hal_iram.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libhal.a:soc_hal.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libhal.a:spi_flash_encrypt_hal_iram.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libhal.a:spi_flash_hal_iram.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libhal.a:spi_hal_iram.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libhal.a:spi_slave_hal_iram.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libhal.a:wdt_hal_iram.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libheap.a:heap_tlsf.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libheap.a:multi_heap.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libnewlib.a:abort.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libnewlib.a:assert.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libnewlib.a:heap.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
    *libnewlib.a:stdatomic.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libesp_system.a:esp_err.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libesp_system.a:ubsan.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libgcov.a:(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libhal.a:cache_hal_esp32.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libhal.a:i2c_hal_iram.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libhal.a:ledc_hal_iram.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libhal.a:mmu_hal.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libhal.a:spi_flash_encrypt_hal_iram.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libhal.a:spi_flash_hal_iram.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libhal.a:spi_hal_iram.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libhal.a:spi_slave_hal_iram.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libhal.a:wdt_hal_iram.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libnewlib.a:abort.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libnewlib.a:assert.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libnewlib.a:heap.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
                                                                    *libnewlib.a:stdatomic.*(.rodata .rodata.* .sdata2 .sdata2.* .srodata .srodata.*)
```