SE Host Services API

Version 1.107.0

SERVICES Library 0.50.7

Table of Contents

Introduction

SERVICES Library Change log

-
-
    -
        -
        -
    -
        -
        -
    -

SERVICES Library Pairings

SERVICES Version	SES Version	NOTES

Known Limitations

A32

SE-UART Spamming

SE Host Services summary

Service Group		Notes
Maintenance		
	SERVICES_heartbeat	
System Management		
	SERVICES_system_get_toc_data	
	SERVICES_system_get_toc_number	
	SERVICES_system_get_toc_via_name	
	SERVICES_system_get_toc_version	
	SERVICES_system_get_toc_via_cpuid	
	SERVICES_system_get_device_part_number	
	SERVICES_system_get_device_data	
	SERVICES_system_set_services_debug	
	SERVICES_system_get_otp_data	
	SERVICES_system_read_otp	
	SERVICES_get_se_revision	
	SERVICES_system_get_eui_extension	
	SERVICES_system_get_ecc_public_key	
Application / Pin mux management		
	SERVICES_pinmux	
	SERVICES_padcontrol	
	SERVICES_uart_write	
	SERVICES_application_ospi_write_key	
	SERVICES_SRAM_retention_config	
Power		

	SERVICES_power_stop_mode_request	
	SERVICES_power_ewic_config	
	SERVICES_power_wakeup_config	
	SERVICES_power_mem_retention_config	
	SERVICES_power_m55_he_vtor_save	
	SERVICES_power_m55_hp_vtor_save	
	SERVICES_power_memory_req	
	SERVICES_global_standby_mode	
	SERVICES_get_run_cfg	
	SERVICES_set_run_cfg	
	SERVICES_get_off_cfg	
	SERVICES_set_off_cfg	
	SERVICES_power_ldo_voltage_control	
	SERVICES_power_dcdc_voltage_control	
	SERVICES_power_setting_configure	
	SERVICES_power_setting_get	
	SERVICES_power_mem_retention_config_raw	
	SERVICES_power_ewic_config_raw	
	SERVICES_power_wakeup_config_raw	
	SERVICES_power_stop_mode_raw_req	
	SERVICES_power_m55_he_vtor_save_raw	
Security /Crypto		
	SERVICES_cryptocell_get_lcs	
	SERVICES_cryptocell_get_rnd	
	SERVICES_cryptocell_mbedtls_aes_init	
	SERVICES_cryptocell_mbedtls_aes_set_key	
	SERVICES_cryptocell_mbedtls_aes_crypt	
	SERVICES_cryptocell_mbedtls_sha_starts	
	SERVICES_cryptocell_mbedtls_sha_process	
	SERVICES_cryptocell_mbedtls_sha_update	
	SERVICES_cryptocell_mbedtls_sha_finish	
	SERVICES_cryptocell_mbedtls_ccm_gcm_set_key	
	SERVICES_cryptocell_mbedtls_ccm_gcm_crypt	
	SERVICES_cryptocell_mbedtls_ccm_gcm_chachapoly_crypt	
	SERVICES_cryptocell_mbedtls_ccm_gcm_poly1305_crypt	
	SERVICES_cryptocell_mbedtls_cmac_init_setkey	
	SERVICES_cryptocell_mbedtls_cmac_update	
	SERVICES_cryptocell_mbedtls_cmac_finish	
	SERVICES_cryptocell_mbedtls_cmac_reset	
Boot		
	SERVICES_boot_process_toc_entry	
	SERVICES_boot_cpu	
	SERVICES_set_vtor	
	SERVICES_boot_reset_cpu	

	SERVICES_boot_release_cpu	
	SERVICES_boot_reset_soc	
Clock		
	SERVICES_clocks_select_osc_source	
	SERVICES_clocks_select_pll_source	
	SERVICES_clocks_enable_clock	
	SERVICES_clocks_set_ES0_frequency	
	SERVICES_clocks_set_ES1_frequency	
	SERVICES_clocks_select_a32_source	
	SERVICES_clocks_select_aclk_source	
	SERVICES_clocks_set_divider	
	SERVICES_clocks_get_clocks	
	SERVICES_clocks_get_apb_frequency	
	SERVICES_clocks_get_refclk_frequency	
	SERVICES_pll_initialize	
	SERVICES_pll_deinit	
	SERVICES_pll_xtal_start	
	SERVICES_pll_xtal_stop	
	SERVICES_pll_xtal_is_started	
	SERVICES_pll_clkpll_start	
	SERVICES_pll_clkpll_stop	
	SERVICES_pll_clkpll_is_locked	
Update		
	SERVICES_update_stoc	

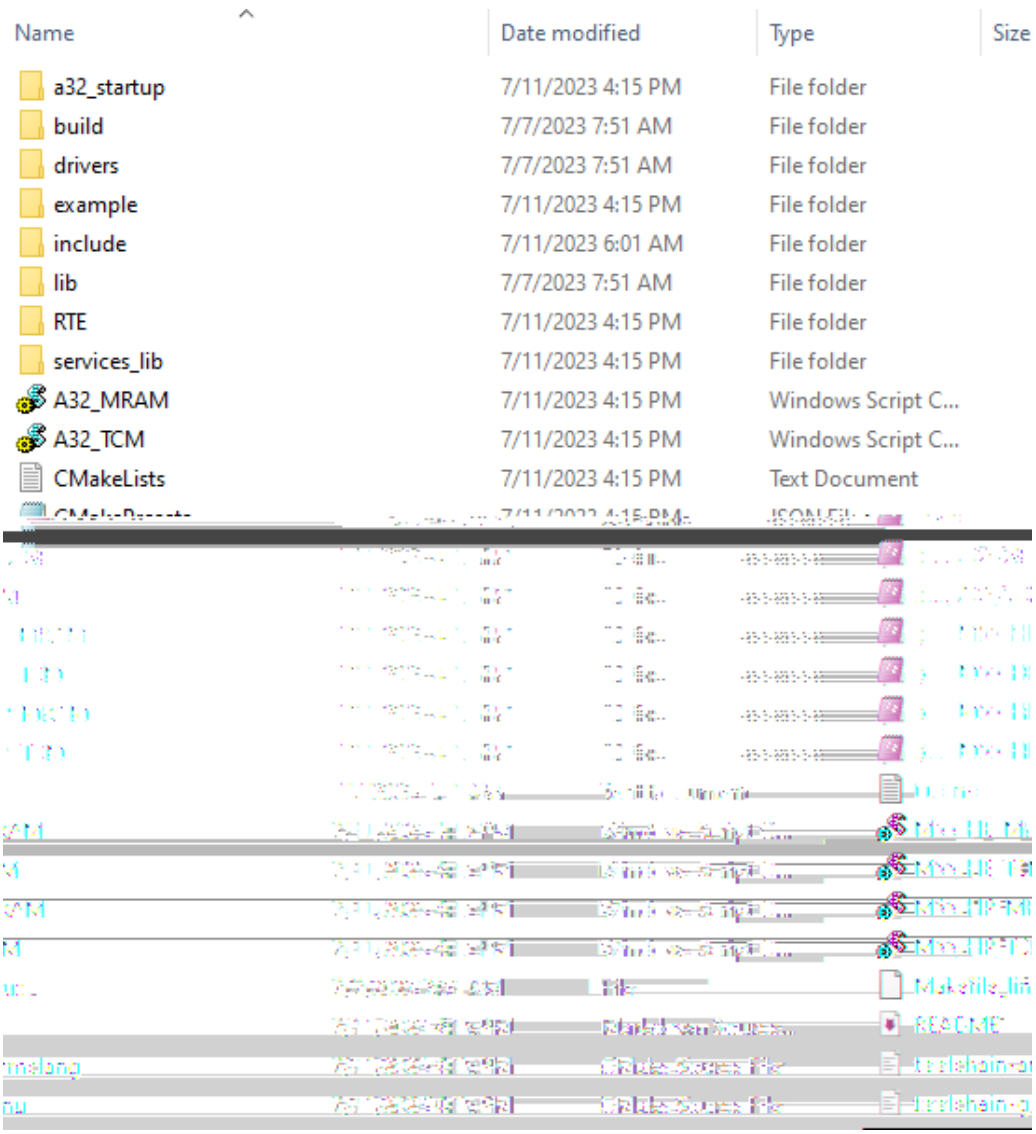
SE Host Services Delivery Components

-
  -
-
-
  -
-
  -
  -
-
  -

Pre-requisites

-
  -
  -
-
-
-
  -

Building SE Services – Windows / LINUX



Name	Purpose	Notes

Building with ARM GNU C

```
$ cd se-host-service-release
$ mkdir build_he_gcc_tcm
$ cd build_he_he_gcc_tcm
$ cmake .. -G "Unix Makefiles" -DENSEMBLE_CORE=M55_HE -
DCMAKE_TOOLCHAIN_FILE=toolchain-gnu.cmake
$ make install

$ cd se-host-service-release
$ mkdir build_he_power_gcc_tcm
$ cd build_he_power_gcc_tcm
$ cmake .. -G "Unix Makefiles" -DENSEMBLE_CORE=M55_HE -
DCMAKE_TOOLCHAIN_FILE=toolchain-gnu.cmake -DPOWER=ON
$ make install
```

Building with ARM CLANG

```
$ cd se-host-services-release
$ mkdir build_he_power_clang_tcm
$ cd build_he_power_clang_tcm
$ cmake .. -G "Unix Makefiles" -DENSEMBLE_CORE=M55_HE -DPOWER=ON
DCMAKE_TOOLCHAIN_FILE=toolchain-armclang.cmake
$ make install -j 8

$ cd se-host-services-release
$ mkdir build_he_clang_tcm
$ cd build_he_clang_tcm
$ cmake .. -G "Unix Makefiles" -DENSEMBLE_CORE=M55_HE -
DCMAKE_TOOLCHAIN_FILE=toolchain-armclang.cmake
$ make install -j 8
```

SERVICES Library Dependencies

-
-

CMSIS Package

JSON Configurations

-
-
-

-
-
-
-
-

JSON File	TCM	MRAM
	✓	
		✓
	✓	
	✓	
	✓	
		✓
	✓	
		✓
	✓	
		✓
	✓	✓

Power Example

BASIC1 (No XIP)

-
-

BASIC2 (XIP)

-
-

BASIC3 (No XIP)

-

BASIC4 (No XIP)

-
    -
    -
    -
    -
-
- **NOTE**: There is an issue when measuring the CPU speeds of both M55s at the same time. It seems it is caused by the shared usage of RTC_A, which is also used for wakeup logic.

BASIC5 (No XIP)

-

BASIC6

-

BASIC7 (No XIP)

-

BASIC8 to BASIC13 Clock configuration examples (No XIP)

-
-

BASIC14 GET request examples (No XIP)

-

BASIC15 Clock Source Cycling (No XIP)

-
-

BASIC16 M55s run in TCM not retained (No XIP)

-
-
-
-
-
-
    -
-
    -
        -
    -
-
    -

BASIC17 PD5=OFF and Wake up

-

BASIC18 Raw power services

-
-
-
-

Power Consumption Examples

-
-
-
-
-
-
-

SES Power Policies

- Logic added to process ATOC and boot M55_HE and M55_HP in case the M55_HE TCM was not retained.
- SES will apply retention settings as soon as a service request is received because they cannot be applied after a subsystem goes OFF.

Power Example Running – Debugging common issues

ISP Not responding

Cannot use UpdateSystemPackage

-

```
$ updatesystempackage -s
$ app-write-mram -s
```

PPU Interrupt Spamming

```
[SES] es0 ppu_isr=0x80
[SES] es0 ppu_aisr=0x2

[SES] es0 PPU PPU_PWRP=0x100 PPU_PWSR=0x108
[SES] es0 ppu_isr=0x80
[SES] es0 ppu_aisr=0x2

[SES] es0 PPU PPU_PWRP=0x100 PPU_PWSR=0x108
[SES] es0 ppu_isr=0x80
[SES] es0 ppu_aisr=0x2

[SES] es0 PPU PPU_PWRP=0x100 PPU_PWSR=0x108
[SES] es0 ppu_isr=0x80
[SES] es0 ppu_aisr=0x2

[SES] es0 PPU PPU_PWRP=0x100 PPU_PWSR=0x108
[SES] es0 ppu_isr=0x80
[SES] es0 ppu_aisr=0x2

[SES] es0 PPU PPU_PWRP=0x100 PPU_PWSR=0x108
[SES] es0 ppu_isr=0x80
[SES] es0 ppu_aisr=0x2

[SES] es0 PPU PPU_PWRP=0x100 PPU_PWSR=0x108
[SES] es0 ppu_isr=0x80
[SES] es0 ppu_aisr=0x2
```

Power Examples Limitations

- Wakeup timers that expire before you enter STOP mode.
  - You will not enter stop mode.
- The booting of specific CPU core as per requested wake up event is not yet supported.
  - On any configured wake up event the m55-he is booted if its TCM is retained or ATOC is processed, and bootable images are booted (potentially both m55-he and m55-hp if the ATOC has bootable images for both)
- Some tests may not report on the SE-UART as the clock rates are too low.

SES Clock Policies

- SES COLD Boot
  - SES checks for presence of Application Conductor objects specifying Clock directions.
  - If no DCT object is present, SES will set the LF Clock Source to the LFXO (Default)

SERVICES test harness example

```
example\common\services_test.c
```

Examples customization options.

```
services_test.c

efg of TEST_PRINT_ENABLE          1   /* Enable printing from Test harness */
efg of PRINT_VIA_CONSOLE          0   /* Print via Debugger console        */
efg of PRINT_VIA_SE_UART          1   /* Print via SE UART terminal        */
```

Flag	Meaning

Changing CMSIS Packs

-
-
-

```
$ cmake .. -G "Unix Makefiles" -DENSEMBLE_CORE=M55_HE -
DCMAKE_TOOLCHAIN_FILE=toolchain-armclang.cmake -Dalifcmsis=”1.3.0”
```

Building and running the Examples

Building the M55 HE Example - run from TCM.

-
    -
-
    -

```
$ cd <host-release directory>
$ mkdir build_he
$ cd build_he
$ cmake .. -G "Unix Makefiles" -DENSEMBLE_CORE=M55_HE -
DCMAKE_TOOLCHAIN_FILE=toolchain-armclang.cmake
$ make install
```

```
$ cd <release directory>/app-release-exec
$ ./app-gen-toc -f build/config/service-he.json
$ ./app-write-mram
```

Building the M55 HE Example - run from MRAM.

-
    -
-
    -

```
$ cd <host-release directory>
$ mkdir build_he_mram
$ cd build_he_mram
$ cmake .. -G "Unix Makefiles" -DENSEMBLE_CORE=M55_HE -
DCMAKE_TOOLCHAIN_FILE=toolchain-armclang.cmake -DXIP=1
$ make install
```

```
$ cd <release directory>/app-release-exec
$ ./app-gen-toc -f build/config/service-he-xip.json
$ ./app-write-mram
```

Building and running the M55_HE Power Example (ARM Clang)

```
$ cd se-host-service-release
$ mkdir build_he_power
$ cd build_he_power
$ cmake .. -G "Unix Makefiles" -DENSEMBLE_CORE=M55_HE -
DCMAKE_TOOLCHAIN_FILE=toolchain-armclang.cmake -DPOWER=ON
$ make install

$ cd ../app-release-exec
$ ./app-gen-toc -f build/config/service-he.json
$ ./app-write-mram
```

Building and running the M55_HE Power Example (ARM GNUC)

```
$ cd se-host-service-release
$ mkdir build_he_power
$ cd build_he_power
$ cmake .. -G "Unix Makefiles" -DENSEMBLE_CORE=M55_HE -
DCMAKE_TOOLCHAIN_FILE=toolchain-gnu.cmake -DPOWER=ON
$ make install

$ cd ../app-release-exec
$ ./app-gen-toc -f build/config/service-he.json
$ ./app-write-mram
```

Building and running the M55_HP Power Example (ARM Clang)

```
$ cd se-host-service-release
$ mkdir build_hp_power
$ cd build_hp_power
$ cmake .. -G "Unix Makefiles" -DENSEMBLE_CORE=M55_HP -
DCMAKE_TOOLCHAIN_FILE=toolchain-armclang.cmake -DPOWER=ON
$ make install

$ cd ../app-release-exec
$ ./app-gen-toc -f build/config/service-hp.json
$ ./app-write-mram
```

Building and running the M55_HP Power Example (ARM GNUC)

```
$ cd se-host-service-release
$ mkdir build_hp_power
$ cd build_hp_power
$ cmake .. -G "Unix Makefiles" -DENSEMBLE_CORE=M55_HP -
DCMAKE_TOOLCHAIN_FILE=toolchain-gnu.cmake -DPOWER=ON
$ make install

$ cd ../app-release-exec
$ ./app-gen-toc -f build/config/service-hp.json
$ ./app-write-mram
```

Building and running the STOC update example

```
$ cd se-host-service-release
$ mkdir stoc_update
$ cd stoc_update
$ cmake .. -G "Unix Makefiles" -DENSEMBLE_CORE=M55_HE -
DCMAKE_TOOLCHAIN_FILE=toolchain-gnu.cmake -DEXAMPLE=UPDATE_STOC
 $ make install
 $ cd <app-release-dir>
 $ <copy the STOC update package to build/images/STOC.bin>
 $ app-gen-toc -f build/config/services-he-update-stoc.json
 $ app-write-mram
```

Building SE Host Services – LINUX

```
$ make -f Makefile_linux lib
```

Installing examples

```
$ cd se-host-services-release
```

```
              INSTALL_DIR
```

```
app-release-exec-windows-SE-FW_0.<version>
      + app-release-exec
            + build
                  + config
                  + images
```

```
$ cd se-host-services-release
$ mkdir build_he
$ cd build_he
$ cmake .. -G "Unix Makefiles" -DENSEMBLE_CORE=M55_HE -
DCMAKE_TOOLCHAIN_FILE=toolchain-armclang.cmake
$ make install -j 8
```

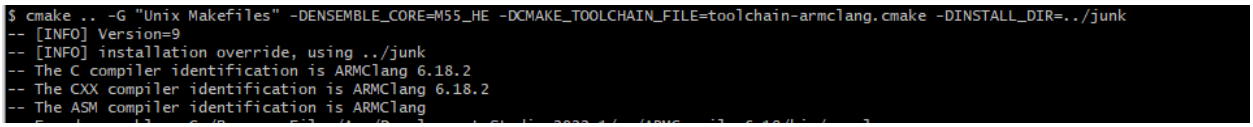
Installing examples to a different location

```
-DINSTALL_DIR=<some path>
```

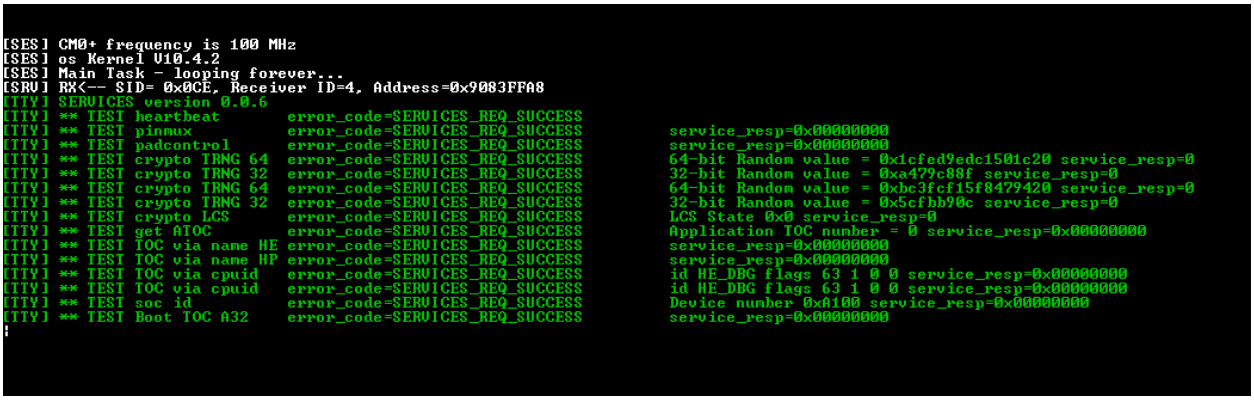
```
                 INSTALL_DIR                app-release-exec
```

```
$ cd se-host-services-release
$ mkdir build_he
```

```
$ cd build_he
$ cmake .. -G "Unix Makefiles" -DENSEMBLE_CORE=M55_HE -
DCMAKE_TOOLCHAIN_FILE=toolchain-armclang.cmake -DINSTALL_DIR=<some
location>
$ make install -j 8
```

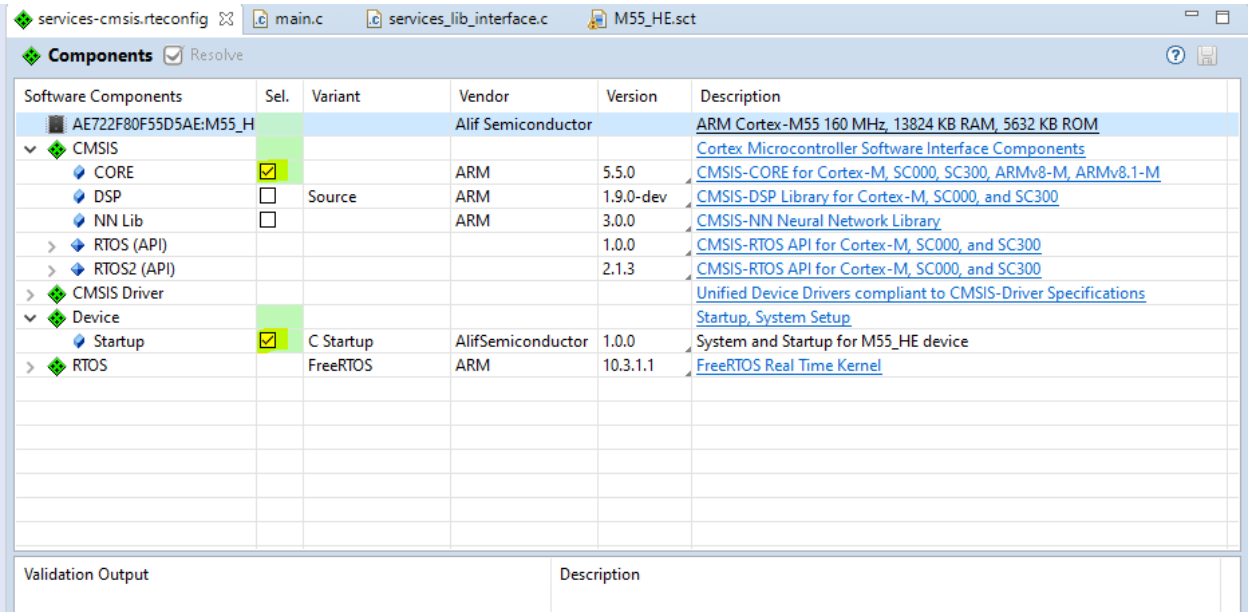


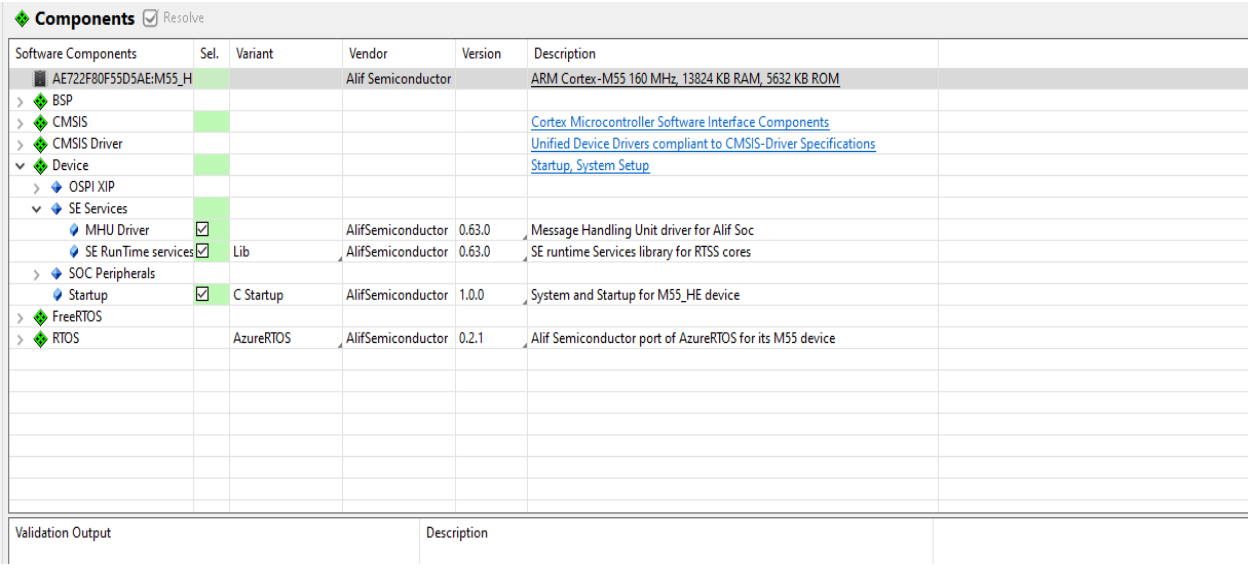
Running with SERVICES Debug disabled



Building the M55 Host Example under ARM-DS

-
-
-
  -
  -





Adding ALIF SE SERVICES to your Application code

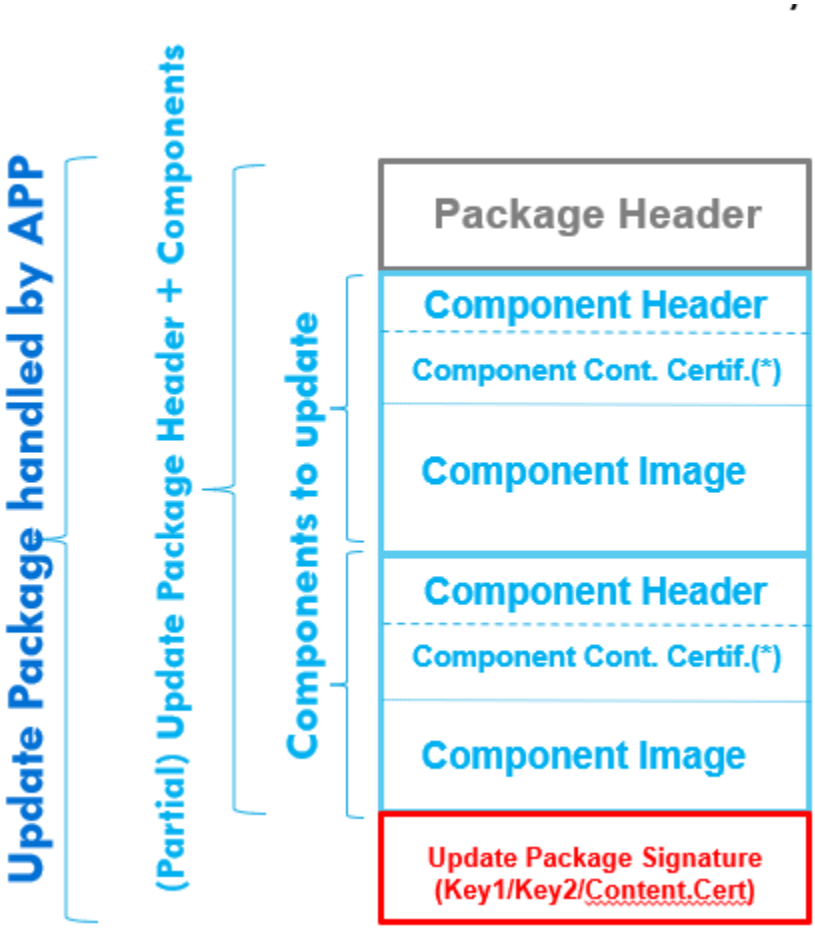
-
  - /service-release/include/services_lib_api.h
  - /service-release/include/aipm.h
-
  - /service-release/lib/libservices_m55_lib.a (or _a32_)
  - /service-release/lib/libmhu_m55_lib.a (or _a32_)
- service_lib_interface.c
  -
  -
  -

SE Firmware OTA Updates

-
-

Update Packages

-
-



Update Packages Header

```
u  fefg tusvdu
{
   uint8_t  header[4];         /*!< identifier; FUPD (Full) or PUPD (Partial) */
   uint16_t version;           /*!< Update version */
   uint16_t header_size;       /*!< UPD Package Header size (in bytes) */
   uint16_t number_of_entries; /*!< Number of components (only for Partial) */
   uint16_t entry_header_size; /*!< Size of entry header (only for Partial) */
   uint32_t package_size       /* to know where the Certs.chain starts */
   uint8_t  device_package[16] /* Device Package to match with Device */
} update_package_header_t;
```

```
typedef struct
{
   uint32_t object_length;     /*!< Length in bytes */
   uint32_t object_pad;        /*!<  pad in bytes */
   uint32_t flags;             /*!< TOC object operators */
   uint32_t version;           /*!< Versioning for this object */
   uint8_t  image_identifier[TOC_ENTRY_IMAGE_NAME_LENGTH];  /*!< Image name*/
   Uint8_t  pad[8]            /* pad with 0s to complete 16-byte
alignment  */
} update_component_header_t;
```

Full Update Packages

Partial Update Packages

Update Packages acceptance policy

-
    -
-
    -
-
    -
-

```
uint32_t SERVICES_update_stoc(uint32_t services_handle,
                              uint32_t image_address,
                              uint32_t image_size,
                              uint32_t *error_code)
```

SE Host Services Library API

-
-
-
-

SERVICES_wait_ms(uint32_t wait_time_ms)	
SERVICES_send_mhu_message_to_se(uint32_t message)	

Host Services Library Interface API Porting Layer

services_lib_interface.c

SERVICES_wait_ms

```
// Delay function
int wait_ms(uint32_t wait_time_ms)
```

SERVICES_send_mhu_message_to_se

```
// MHU send message to SE on MHU0 channel0
int send_mhu_message_to_se(uint32_t message)
```

Host Services Library API Layer

SERVICES_initialize

```
// Service library initialization
void SERVICES_initialize(services_lib_t *init_params)

SERVICES_initialize(services_lib_t *init_params);

// Service synchronization
int SERVICES_synchronize_with_se(uint32_t services_handle)

number_of_retries = SERVICES_synchronize_with_se(services_handle);
```

SERVICES_send_request

```
// Service request call
uint32_t SERVICES_send_request(uint32_t services_handle,
                               uint16_t service_id,
                               uint32_t service_timeout);

Error_code = SERVICES_send_request(handle, SERVICE_HEARTBEAT_ID,
DEFAULT_TIMEOUT);
```

SERVICES_send_msg_acked_callback

```
// MHU message ACK callback function
void SERVICES_send_msg_acked_callback(void)
```

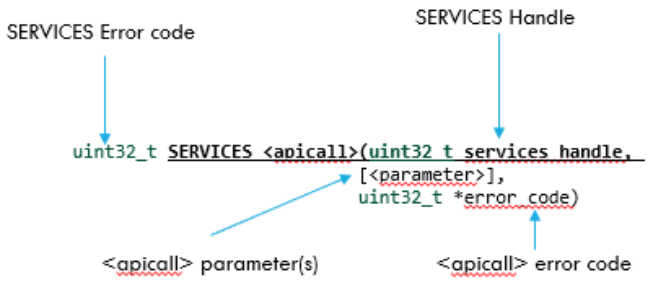
SERVICES_rx_msg_callback

```
// MHU message received callback function
void SERVICES_rx_msg_callback(uint32_t message);
```

```
// Pinmux service
int PINMUX_config(Port_t port_num, Pin_t pin_num, Pinfunction_t
function);
```

SE Host SERVICES Library

SE Host SERVICES Library - Anatomy of a SERVICE Call



-
-
    -
-
    -
    -
    -

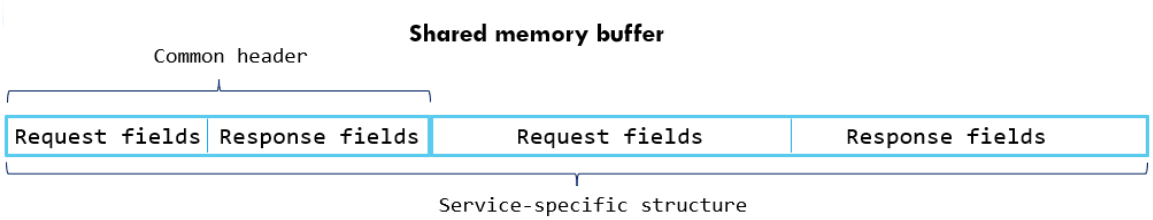
SE Host Service Library Internal implementation

- `example_service_t`
-
-
-

-
-

```c
typedef struct
{
  service_header_t header;
  volatile uint32_t send_<param>;      /*!< Send   parameter */
  volatile uint32_t resp_<param>;      /*!< Return parameter */
  volatile uint32_t resp_error_code;   /*!< Call error code  */
} example_service_t;

void SERVICES_example_call(services_req_t *service)
{
  example_service_t *p_svc =
    (example_service_t *)service->pkt_buffer_address; /* services request */
  uint32_t error_code;
  uint32_t local_result;

  error_code = function_call(p_svc->send_<param>, &local_result);

  p_svc->resp_<param>    = local_result;
  p_svc->resp_error_code = error_code;

  SERVICES_send_response_code(service, SERVICES_REQ_SUCCESS);
}
```

SE Host Service Library Transport Protocol details



Common header format

Service ID: req
Flags: req
Error Code: resp
Padding

```c
typedef struct
{
  uint16_t send_sid;
  uint16_t send_flags;
  uint16_t resp_error_code; // transport
layer error code
  uint16_t none_padding;
} service_header_t;
```

Service-specific structure example

```c
typedef struct
{
  service_header_t header;
  uint8_t send_port_num;
  uint8_t send_pin_num;
  uint8_t send_config_data;
  uint8_t resp_error_code; // service-specific
error code
} pinmux_svc_t;
```

SE Host Service Library Transport Error Codes

SERVICES_REQ_SUCCESS		
SERVICES_REQ_NOT_ACKNOWLEDGE		
SERVICES_REQ_TIMEOUT		
SERVICES_REQ_UNKNOWN_COMMAND		

SE Host Services Library Error Handling

-
-

```
efg of SERVICES_REQ_SUCCESS                    0x00
efg of SERVICES_REQ_NOT_ACKNOWLEDGE            0xFF
efg of SERVICES_REQ_TIMEOUT                    0xFD
efg of SERVICES_RESP_UNKNOWN_COMMAND           0xFC
efg of SERVICE_SUCCESS                         0x0
efg of SERVICE_FAIL                            0x200
efg of SERVICE_INVALID_PARAMETER               0x201
```

SE Host Services Library Memory handling

SE Host Services API

Miscellaneous

SERVICES_Initialize

Syntax:

Description:

-
-
  -
  -
-
-
-

Parameters:

Returns:

Restrictions:

Example:

```
#include “services_lib_api.h” /* services_lib_t lives here */

tu u d uint8_t
  s_packet_buffer[SERVICES_MAX_PACKET_BUFFER_SIZE]    uus cvuf   ((aligned (4)));

 ou          s ou(dpotu di s * fmt, ...)
{
  /* To be filled in by the user */

  sfuvso 0;
}
```

```
int32_t          x  u nt(uint32_t wait_time_ms)
{
  /* To be filled in by the user */

  sfuvso 0;
}

int main (void)
{
  uint32_t ErrorCode = SERVICES_OK;

  services_lib_t  services_init_params =
  {
    .global_offset         = DTCM_GLOBAL_ADDRESS - M55_DTCM_LOCAL_OFFSET,
    .packet_buffer_address = (uint32_t)s_packet_buffer,
    .fn_send_mhu_message   = send_message,
    .fn_wait_ms            = &SERVICES_wait_ms,
    .wait_timeout          = timeout,
    .fn_print_msg          = &SERVICES_print,
  };

  ErrorCode = SERVICES_initialize(&services_init_params);

   g (ErrorCode != SERVICES_REQ_SUCCESS)
  {
    sfuvso ErrorCode;
  }
}
```

SERVICES_version

Syntax:

Description:

Parameters:

Returns:

Restrictions:

Example:

```
#include <services_lib_api.h>

int main (void)
{
  uint32_t ErrorCode = SERVICES_OK;

  printf("SERVICES version %s\n", SERVICES_version());
}
```

SERVICES_register_channel

Syntax:

Description:

Parameters:

Returns:

Restrictions:

Example:

```
#include <services_lib_api.h>

 efg of NUM_MHU                2

 efg of MHU_M55_SE_MHU0        0
 efg of MHU_M55_SE_MHU1        1

int main (void)
{
  mhu_initialize();
  SERVICES_Setup(s_mhu_driver_out.send_message, MAXIMUM_TIMEOUT);

  uint32_t services_handle = SERVICES_register_channel(MHU_M55_SE_MHU0, 0);

  printf("SERVICES handle %d\n", services_handle);
}
```

SERVICES_prepare_packet_buffer

Syntax:

Description:

Parameters:

Returns:

Restrictions:

Example:

SERVICES_heartbeat

Syntax:

Description:

Parameters:

Returns:

Restrictions:

Example:

```
int main (void)
{
  uint32_t ErrorCode = SERVICES_OK;

  mhu_initialize();
  SERVICES_Setup(s_mhu_driver_out.send_message, MAXIMUM_TIMEOUT);

  //SERVICES_wait_ms(0x1000000);

  uint32_t services_handle = SERVICES_register_channel(MHU_M55_SE_MHU0, 0);

  ErrorCode = SERVICES_heartbeat(              );
   g (ErrorCode != SERVICES_REQ_SUCCESS)
  {
    sfuvso ErrorCode;
  }

  sfuvso ErrorCode;

}
```

SERVICES_synchronize_with_se

Syntax:

Description:

Parameters:

Returns:

Restrictions:

Example:

```
int main (void)
{
   ou retry_count;

  /* keep sending heartbeat services requests until one succeeds */
  retry_count = SERVICES_synchronize_with_se(services_handle);

}
```

SERVICES_system_set_services_debug

Syntax:

```
uint32_t          t tufn tfu tfsw dft efcvh (uint32_t services_handle,
                                             bool debug_enable,
                                             uint32_t *error_code)
```

Description:

Parameters:

Returns:

Restrictions:

Example:

```
int main (void)
{
  uint32_t service_error_code;

  SERVICES_system_set_services_debug(services_handle,
                                     false,/* False = NO debug output */
                                     &service_error_code);
  if (service_error_code != SERVICES_REQ_SUCCESS)
  {
     /* Deal with error */
  }
}
```

SERVICES_system_read_otp

Syntax:

```
uint32_t          t tufn sf e pu      (uint32_t services_handle,
                                       uint32_t otp_offset,
                                       uint32_t *otp_value_word,
                                       uint32_t *error_code)
```

Description:

Parameters:

Returns:

Restrictions:

Example:

```
int main (void)
{
  uint32_t ErrorCode = SERVICES_REQ_SUCCESS
  uint32_t service_error_code;
  uint32_t otp_value;

  ErrorCode = SERVICES_system_read_otp(services_handle,
                                       0x70, /* Offset */
                                       &otp_value,
                                       &service_error_code);
   g (ErrorCode != SERVICES_REQ_SUCCESS)
  {
    sfuvso ErrorCode;
  }
}
```

SERVICES_system_write_otp

Syntax:

```
uint32_t          t tufn xs uf pu      (uint32_t services_handle,
                                        uint32_t otp_offset,
                                        uint32_t otp_value_word,
                                        uint32_t *error_code)
```

Description:

Parameters:

Returns:

SERVICES_system_get_otp_data

Syntax:

```
uint32_t          t tufn hfu pu  e u  (uint32_t services_handle,
                                       SERVICES_otp_data_t *otp_info,
                                       uint32_t * error_code)
```

Description:

Parameters:

Returns:

Restrictions: This function is deprecated. SES does not process this SERVICE command.

Example:

```
int main (void)
{
  uint32_t ErrorCode = SERVICES_OK;
  uint32_t service_error_code;
  SERVICES_otp_data_t otp_info;

  ErrorCode = SERVICES_system_get_otp_data(services_handle,
                                           &otp_info,
                                           &service_error_code);
   g (ErrorCode != SERVICES_REQ_SUCCESS)
  {
    sfuvso ErrorCode;
  }
}
```

SERVICES_system_get_toc_data

Syntax:

```
uint32_t          t tufn hfu upd e u  (uint32_t services_handle,
                                       SERVICES_toc_data_t *toc_info,
                                       uint32_t * error_code)
```

Description:

```
u  fefg tusvdu {
      uint8_t   image_identifier[TOC_NAME_LENGTH]; /*!< TOC name         */
      uint32_t  version;                           /*!< TOC Version      */
      uint32_t  cpu;                               /*!< TOC Cpu ID       */
      uint32_t  store_address;                     /*!< TOC MRAM address */
      uint32_t  load_address;                      /*!< TOC load         */
      uint32_t  boot_address;                      /*!< TOC boot address */
      uint32_t  image_size;                        /*!< TOC image size   */
      uint32_t  processing_time;                   /*!< TOC process time */
      uint32_t  flags;                             /*!< TOC flag state   */
      uint8_t   flags_string[FLAG_STRING_SIZE];    /*!< TOC flag string  */
} SERVICES_toc_info_t;


/**
 * @struct SERVICES_toc_data_t
 */
u  fefg tusvdu
{
  uint32_t number_of_toc_entries;
```

```
  SERVICES_toc_info_t toc_entry[SERVICES_NUMBER_OF_TOC_ENTRIES];
} SERVICES_toc_data_t;
```

Parameters:

Returns:

Restrictions:

Example:

```
int main (void)
{
  uint32_t ErrorCode = SERVICES_OK;
  uint32_t service_error_code;
SERVICES_toc_data_t toc_info;

  ErrorCode = SERVICES_system_get_toc_data(services_handle,
                                           &toc_info,
                                           &service_error_code);
   g (ErrorCode != SERVICES_REQ_SUCCESS)
  {
    sfuvso ErrorCode;
  }
}
```

```
[TTY] ** TEST TOC get data     error_code=SERVICES_REQ_SUCCESS          TOC number = 5 service_resp=0x00000000
[TTY] +-------------------------------------------------------------------------------+
[TTY] |    Name    |     CPU    |Load Address|Boot Address|Image Size|  Version  | Flags |
[TTY] +-------------------------------------------------------------------------------+
[TTY] |    DEVICE  |  CM0+      | 0x00000000 | 0x00000000 |      296 |      0.5.0| u V   |
[TTY] |    DEVICE  |  CM0+      | 0x00000000 | 0x00000000 |      372 |      0.5.0| u V   |
[TTY] |    SERAM0  |  CM0+      | 0x00000000 | 0x00000000 |    57612 |    1.105.0| u s   |
[TTY] |    SERAM1  |  CM0+      | 0x00000000 | 0x00000000 |    57612 |    1.105.0| u s   |
[TTY] | SRV-HE-T   |  M55_HE    | 0x58000000 | 0x58000000 |    73440 |      1.0.0| uLVB  |
[TTY] +-------------------------------------------------------------------------------+
```

SERVICES_system_get_toc_number

Syntax:

```
uint32_t          t tufn hfu upd ovncfs(uint32_t services_handle,
                                        uint32_t *toc_number,
                                        uint32_t * error_code)
```

Description:

Parameters:

Returns:

Restrictions:

Example:

```
int main (void)
{
  uint32_t ErrorCode = SERVICES_OK;
  uint32_t number_of_tocs;
  uint32_t service_error_code;

  ErrorCode = SERVICES_system_get_toc_number(services_handle,
                                             &number_of_tocs,
                                             &service_error_code);
   g (ErrorCode != SERVICES_REQ_SUCCESS)
  {
    sfuvso ErrorCode;
  }
}
```

SERVICES_system_get_toc_version

Syntax:

```
uint32_t          t tufn hfu upd wfst po (uint32_t services_handle,
                                          uint32_t *toc_version,
                                          uint32_t *error_code)
```

Description:

```
/* Unpack the SE version */
major = (version >> 24) & 0xFF;
minor = (version >> 16) & 0xFF;
patch = (version >>  8) & 0xFF;
```

Parameters:

Returns:

SERVICES_SUCCESS
toc_version

Restrictions:

SERVICES_system_get_toc_via_name

Syntax:

```
uint32_t          t tufn hfu upd w   o nf(uint32_t services_handle,
                                          dpotu uint8_t *cpu_name,
                                          uint32_t * error_code);
```

Description:

Parameters:

Returns:

Restrictions:

Example:

SERVICES_system_get_toc_via_cpuid

Syntax:

```
uint32_t          t tufn hfu upd w   d v e(uint32_t services_handle,
                                           SERVICE_cpuid_t cpuid,
                                           SERVICES_toc_data_t *toc_info,
                                           uint32_t * error_code);
```

Description:

```
u  fefg fovn {
      HOST_CPU_0    = 0,                /**< A32_0 CPU                */
      HOST_CPU_1    = 1,                /**< A32_1 CPU                */
      EXTSYS_0      = 2,                /**< M55 HP CPU or other CPU */
      EXTSYS_1      = 3,                /**< M55 HE CPU               */
} SERVICES_cpuid_t;
```

Parameters:

Returns:

Restrictions:

Example:

```
int main (void)
{
  uint32_t ErrorCode = SERVICES_OK;
  SERVICES_toc_data_t toc_info;
  Uint32_t service_error_code;

 error_code = SERVICES_system_get_toc_via_cpuid(services_handle,
                                                FUSION_M55_HE,
                                                &toc_info,
                                                &service_error_code);
   g (ErrorCode != SERVICES_REQ_SUCCESS)
  {
    sfuvso ErrorCode;
```

```
  }

  /* Process each TOC entry found */
  for (int each_toc = 0; each_toc < toc_info.number_of_toc_entries; each_toc++)
  {
    SERVICES_toc_info_t *toc_entry_p;

    toc_entry_p = (SERVICES_toc_info_t *)&toc_info.toc_entry[each_toc];

    /* do something with the TOC information */
}
```

SERVICES_system_get_device_part_number

Syntax:

```
uint32_t          t tufn hfu efw df   su ovncfs(uint32_t services_handle,
                                                uint32_t *device_part_number,
                                                uint32_t *error_code)
```

Description:

Parameters:

Returns:

0x0000B200

Restrictions:

Example:

```
int main (void)
{
  uint32_t ErrorCode = SERVICES_OK;
  uint32_t device_id;
  uint32_t service_error_code;

  ErrorCode = SERVICES_system_get_device_part_number(services_handle,
                                                     &device_part_number,
                                                     &service_error_code);
   g (ErrorCode != SERVICES_REQ_SUCCESS)
  {
    sfuvso ErrorCode;
  }
}
```

SERVICES_system_get_device_data

Syntax:

```
uint32_t          t tufn hfu efw df e u (uint32_t services_handle,
                                        SERVICES_version_data_t *device_info,
                                        uint32_t *error_code)
```

Description:

```
u  fefg tusvdu {
       uint32_t revision_id; /*!< SoC revision          */
       uint8_t version[4];   /*!< @todo deprecate       */
       uint8_t ALIF_PN[16];  /*!< SoC part number       */
       uint8_t HBK0[16];     /*!< ALIF Key              */
       uint8_t HBK1[16];     /*!< ALIF Key              */
       uint8_t HBK_FW[20];   /*!< ALIF Firmware version */
       uint8_t config[4];    /*!< Wounding data         */
       uint8_t DCU[16];      /*!< DCU settings          */
       uint8_t MfgData[32];  /*!< Manufacturing data    */
       uint8_t SerialN[8];   /*!< SoC Serial number     */
       uint8_t LCS;          /*!< SoC lifecycle state   */
       uint32_t external_config[4]; /*!< External mem   */
       uint32_t flags2;      /*!< Alt path options      */
} SERVICES_version_data_t;
```

Parameters:

Returns:

Restrictions:

Example:

```
int main (void)
{
  uint32_t ErrorCode = SERVICES_OK;
  uint32_t device_id;
  SERVICES_version_data_t device_data;
  uint32_t service_error_code;

  ErrorCode = SERVICES_system_get_device_info(services_handle,
```

```
                                                    &device_data,
                                                    &service_error_code);
   g (ErrorCode != SERVICES_REQ_SUCCESS)
  {
    sfuvso ErrorCode;
  }
}
```

SERVICES_get_se_revision

Syntax:

```
uint32_t          hfu tf sfw t po(uint32_t services_handle,
                                  uint8_t *revision_data,
                                  uint32_t *error_code)
```

Description:

Parameters:

Returns:

Restrictions:

Example:

```
int main (void)
{
  uint32_t error_code = SERVICES_REQ_SUCCESS;
  uint32_t service_error_code;
  uint8_t se_revision[80];

  error_code = SERVICES_get_se_revision(services_handle,
                                        (uint8_t*)&se_revision[0],
                                        &services_error_code);

  if (error_code != SERVICES_REQ_SUCCESS)
  {
      /* deal with error */
  }
}
```

SERVICES_system_get_eui_extension

Syntax:

```
uint32_t          t tufn hfu fv  fyufot po(uint32_t services_handle,
                                          bool is_eui48,
                                          uint8_t *eui_extension,
                                          uint32_t *error_code)
```

Description:

Parameters:

SERVICES_system_get_device_id64

Syntax:

```
uint32_t          t tufn hfu efw df  e  (uint32_t services_handle,
                                          uint8_t *device_id,
                                          uint32_t *error_code)
```

Description:

Parameters:

SERVICES_system_get_ecc_public_key

Syntax:

```
uint32_t          t tufn hfu fdd  vcm d lf (uint32_t services_handle,
                                            uint8_t *ecc_pubkey_buffer,
                                            uint32_t *error_code)
```

Description:

Parameters:

Application Services

SERVICES_uart_write

Syntax:

Description:

Parameters:

None

Returns:

Restrictions:

Example:

```
int main (void)
{
  uint32_t ErrorCode = SERVICES_OK;
  uint8_t buffer[256];

  ... <format print buffer>

  ErrorCode = SERVICES_uart_write(services_handle,
                                  sizeof(buffer),
                                  (uint8_t *)buffer);
   g (ErrorCode != SERVICES_REQ_SUCCESS)
  {
    sfuvso ErrorCode;
  }
}
```

SERVICES_pinmux

Syntax:

Description:

Parameters:

Returns:

Restrictions:

Example:

```
int main (void)
{
  uint32_t ErrorCode = SERVICES_OK;
  uint32_t service_error_code;

  ErrorCode = SERVICES_pinmux(              , 1, 14, 0, &service_error_code);
   g (ErrorCode != SERVICES_REQ_SUCCESS)
  {
    sfuvso ErrorCode;
  }
}
```

SERVICES_padcontrol

Syntax:

Description:

Parameters:

Returns:

Restrictions:

Example:

```
int main (void)
{
  uint32_t ErrorCode = SERVICES_OK;
  uint32_t service_error_code;

  ErrorCode = SERVICES_padcontrol(              , 1, 14, 0, &service_error_code);
   g (ErrorCode != SERVICES_REQ_SUCCESS)
  {
    sfuvso ErrorCode;
  }
}
```

SERVICES_application_ospi_write_key

Syntax:

Description:

```
#define OSPI_WRITE_OTP_KEY_OSPI0              0
#define OSPI_WRITE_OTP_KEY_OSPI1              1
#define OSPI_WRITE_EXTERNAL_KEY_OSPI0         2
#define OSPI_WRITE_EXTERNAL_KEY_OSPI1         3
```

Parameters:

Returns:

Restrictions:

Example:

SERVICES_power_stop_mode_request

Syntax:

Description:

Parameters:

Returns:
ErrorCode - SERVICES_REQ_SUCCESS, SERVICES_REQ_CANNOT_EXECUTE_SERVICE

Restrictions:

Example:

```
int main (void)
{
  uint32_t ErrorCode = SERVICES_OK;

  error_code = SERVICES_power_stop_mode_request(            );
   g (ErrorCode != SERVICES_REQ_SUCCESS)
  {
    sfuvso ErrorCode;
  }
}
```

SERVICES_power_ewic_config

Syntax:

Description:

Parameters:

Returns:
ErrorCode - SERVICES_REQ_SUCCESS, SERVICES_REQ_CANNOT_EXECUTE_SERVICE

Restrictions:

Example:

```
int main (void)
{
  uint32_t error_code = SERVICES_REQ_SUCCESS;
  uint32_t ewic_config;

  ewic_config &= (1 << 6);
  error_code = SERVICES_power_ewic_config(services_handle,
                                          ewic_config);

   g (error_code != SERVICES_REQ_SUCCESS)
  {
    sfuvso error_code;
  }
}
```

SERVICES_power_wakeup_config

Syntax:

Description:

Parameters:

```
u  fefg fovn
{
  VBAT_WAKEUP_MDM           = 0x1,          // bit0
  VBAT_WAKEUP_RTC_SE        = 0x10,         // bit4
  VBAT_WAKEUP_RTC_A         = 0x20,         // bit5
  VBAT_WAKEUP_LPCMP         = 0x40,         // bit6
  VBAT_WAKEUP_BROWN_OUT     = 0x80,         // bit7
  VBAT_WAKEUP_LPTIMER       = 0XF00,        // bit11:8
  VBAT_WAKEUP_LPGPIO        = 0XFF0000,     // bit23:16
} SERVICES_wakeup_cfg_t;
```

```
u  fefg fovn
{
  LOWEST_POWER_PROFILE = 0,      /**< LOWEST_POWER_PROFILE */
  HIGH_PERFORMANCE_POWER_PROFILE,/**< HIGH_PERFORMANCE_POWER_PROFILE */
  USER_SPECIFIED_PROFILE,        /**< USER_SPECIFIED_PROFILE */
  DEFAULT_POWER_PROFILE,         /**< DEFAULT_POWER_PROFILE */
  NUMBER_OF_POWER_PROFILES       /**< NUMBER_OF_POWER_PROFILES */
} services_power_profile_t;
```

Returns:
ErrorCode - SERVICES_REQ_SUCCESS, SERVICES_REQ_CANNOT_EXECUTE_SERVICE

Restrictions:

Example:

```
int main (void)
{
```

```
  uint32_t error_code = SERVICES_REQ_SUCCESS;

  error_code = SERVICES_power_wakeup_config(services_handle,
                                            VBAT_WAKEUP_RTC_SE
                                            | VBAT_WAKEUP_RTC_A,
                                            LOWEST_POWER_PROFILE);

   g (error_code != SERVICES_REQ_SUCCESS)
  {
    sfuvso error_code;
  }
}
```

SERVICES_power_mem_retention_config

Syntax:

Description:

Parameters:

```
// Memory retention bit encoding for mem_retention_enable
 efg of POWER_MEM_RET_FIREWALL_RAM             0x01UL
 efg of POWER_MEM_RET_SE_SRAM                  0x02UL
 efg of POWER_MEM_RET_BACKUP_RAM_4KB           0x04UL
// M55-HE TCM RET1: ITCM 0-128kb; DTCM 0-128kb
 efg of POWER_MEM_RET_ES1_TCM_RET1             0x08UL
// M55-HE TCM RET1: ITCM 128-256kb; DTCM 128-256kb
 efg of POWER_MEM_RET_ES1_TCM_RET2             0x10UL
// XTENSA TCM RET1: ITCM 128-512kb
 efg of POWER_MEM_RET_XTENSA_TCM_RET1          0x20UL
// XTENSA TCM RET1: ITCM 64-128kb
 efg of POWER_MEM_RET_XTENSA_TCM_RET2          0x40UL
// XTENSA TCM RET1: ITCM 0-64kb
 efg of POWER_MEM_RET_XTENSA_TCM_RET3          0x80UL
// M55-M TCM RET1: ITCM 1MB; DTCM 384kb
 efg of POWER_MEM_RET_M55_M_TCM_RET1           0x100UL
 efg of POWER_MEM_RET_MODEM_BACKUP_RAM_16KB    0x200UL


u  fefg fovn
{
  LOWEST_POWER_PROFILE = 0,      /**< LOWEST_POWER_PROFILE */
  HIGH_PERFORMANCE_POWER_PROFILE,/**< HIGH_PERFORMANCE_POWER_PROFILE */
  USER_SPECIFIED_PROFILE,        /**< USER_SPECIFIED_PROFILE */
  DEFAULT_POWER_PROFILE,         /**< DEFAULT_POWER_PROFILE */
  NUMBER_OF_POWER_PROFILES       /**< NUMBER_OF_POWER_PROFILES */
} services_power_profile_t;
```

Returns:
ErrorCode - SERVICES_REQ_SUCCESS, SERVICES_REQ_CANNOT_EXECUTE_SERVICE

Restrictions:

Example:
```
int main (void)
{
  uint32_t error_code = SERVICES_REQ_SUCCESS;

  error_code = SERVICES_power_mem_retention_config(services_handle,
                                                   POWER_MEM_RETENTION_SE_RAM,
                                                   LOWEST_POWER_PROFILE);

   g (error_code != SERVICES_REQ_SUCCESS)
  {
    sfuvso error_code;
  }
}
```

SERVICES_power_m55_he_vtor_save

Syntax:

Description:

Parameters:

```
u  fefg fovn
{
  LOWEST_POWER_PROFILE = 0,       /**< LOWEST_POWER_PROFILE */
  HIGH_PERFORMANCE_POWER_PROFILE,/**< HIGH_PERFORMANCE_POWER_PROFILE */
  USER_SPECIFIED_PROFILE,         /**< USER_SPECIFIED_PROFILE */
  DEFAULT_POWER_PROFILE,          /**< DEFAULT_POWER_PROFILE */
  NUMBER_OF_POWER_PROFILES        /**< NUMBER_OF_POWER_PROFILES */
} services_power_profile_t;
```

Returns:

ErrorCode - SERVICES_REQ_SUCCESS, SERVICES_REQ_CANNOT_EXECUTE_SERVICE

Restrictions:

Example:

```
int main (void)
{
  uint32_t error_code = SERVICES_REQ_SUCCESS;

    error_code = SERVICES_power_m55_he_vtor_save(services_handle,
                                                 0x0,
                                                 0x0,
                                                 LOWEST_POWER_PROFILE);
```

```
  g (error_code != SERVICES_REQ_SUCCESS)
  {
    sfuvso error_code;
  }
}
```

SERVICES_power_m55_hp_vtor_save

Syntax:

Description:

Parameters:

```
u  fefg fovn
{
  LOWEST_POWER_PROFILE = 0,       /**< LOWEST_POWER_PROFILE */
  HIGH_PERFORMANCE_POWER_PROFILE,/**< HIGH_PERFORMANCE_POWER_PROFILE */
  USER_SPECIFIED_PROFILE,         /**< USER_SPECIFIED_PROFILE */
  DEFAULT_POWER_PROFILE,          /**< DEFAULT_POWER_PROFILE */
  NUMBER_OF_POWER_PROFILES        /**< NUMBER_OF_POWER_PROFILES */
} services_power_profile_t;
```

Returns:

ErrorCode - SERVICES_REQ_SUCCESS, SERVICES_REQ_CANNOT_EXECUTE_SERVICE

Restrictions:

Example:

```
int main (void)
{
  uint32_t error_code = SERVICES_REQ_SUCCESS;

    error_code = SERVICES_power_m55_hp_vtor_save(services_handle,
                                                 0x0,
                                                 0x0,
                                                 LOWEST_POWER_PROFILE);
```

```
  g (error_code != SERVICES_REQ_SUCCESS)
  {
    sfuvso error_code;
  }
}
```

SERVICES_corestone_standby_mode

Syntax:

Description:

Parameters:

Returns:
ErrorCode - SERVICES_REQ_SUCCESS, SERVICES_REQ_CANNOT_EXECUTE_SERVICE

Restrictions:

Example:

```
int main (void)
{
  uint32_t error_code = SERVICES_REQ_SUCCESS;
    host_cpu_clus_pwr_req_t host_cpu_clus_pwr_req;
    bsys_pwr_req_t bsys_pwr_req;

    host_cpu_clus_pwr_req.word = 0;
    host_cpu_clus_pwr_req.bits.mem_ret_req = 0;
    host_cpu_clus_pwr_req.bits.pwr_req = 1;

    bsys_pwr_req.word = 0;
    bsys_pwr_req.bits.systop_pwr_req = 1;
    bsys_pwr_req.bits.dbgtop_pwr_req = 0;
    bsys_pwr_req.bits.refclk_req = 1;
    bsys_pwr_req.bits.wakeup_en = 0;


    error_code = SERVICES_corestone_standby_mode(services_handle,
                                                 host_cpu_clus_pwr_req,
```

```
                                                bsys_pwr_req,
                                                &service_error_code);

   g (error_code != SERVICES_REQ_SUCCESS)
  {
    sfuvso error_code;
  }
}
```

SERVICES_power_memory_req

Syntax:

Description:

```
POWER_MEM_SRAM_0_ENABLE
POWER_MEM_SRAM_1_ENABLE
POWER_MEM_SRAM_0_ISOLATION_ENABLE
POWER_MEM_SRAM_1_ISOLATION_ENABLE
POWER_MEM_MRAM_ENABLE
```

NOTE

Parameters:

Returns:
ErrorCode - SERVICES_REQ_SUCCESS, SERVICES_REQ_CANNOT_EXECUTE_SERVICE

Restrictions:

Example:

```
int main (void)
{
  error_code = SERVICES_power_memory_req(services_handle,
                                         (POWER_MEM_SRAM_0_ENABLE |
                                          POWER_MEM_SRAM_1_ISOLATION_ENABLE),
                                         &return_error_code);


   g (error_code != SERVICES_REQ_SUCCESS)
  {
    sfuvso error_code;
  }
}
```

SERVICES_get_run_cfg

Syntax:

Description:

Parameters:

Returns:
ErrorCode - SERVICES_REQ_SUCCESS, SERVICES_REQ_CANNOT_EXECUTE_SERVICE

Restrictions:

Example:

```
int main (void)
{
  uint32_t error_code = SERVICES_REQ_SUCCESS;
  run_profile_t runp;

  error_code = SERVICES_get_run_cfg(services_handle, &runp,
                                    &service_error_code);

   g (error_code != SERVICES_REQ_SUCCESS)
  {
    sfuvso error_code;
  }
}
```

SERVICES_set_run_cfg

Syntax:

Description:

Parameters:

Returns:
ErrorCode - SERVICES_REQ_SUCCESS, SERVICES_REQ_CANNOT_EXECUTE_SERVICE

Restrictions:

Example:

```
int main (void)
{
  uint32_t error_code = SERVICES_REQ_SUCCESS;
  run_profile_t runp;

  error_code = SERVICES_set_run_cfg(services_handle, &runp,
                                    &service_error_code);


   g (error_code != SERVICES_REQ_SUCCESS)
  {
    sfuvso error_code;
  }
}
```

SERVICES_get_off_cfg

Syntax:

Description:

Parameters:

Returns:
ErrorCode - SERVICES_REQ_SUCCESS, SERVICES_REQ_CANNOT_EXECUTE_SERVICE

Restrictions:

Example:

```
int main (void)
{
  uint32_t error_code = SERVICES_REQ_SUCCESS;
                                            ewic_config);
  off_profile_t off_mode;

  error_code = SERVICES_set_off_cfg(services_handle,
                                    &off_mode,
                                    &service_error_code);


   g (error_code != SERVICES_REQ_SUCCESS)
  {
    sfuvso error_code;
  }
}
```

SERVICES_set_off_cfg

Syntax:

Description:

Parameters:

Returns:
ErrorCode - SERVICES_REQ_SUCCESS, SERVICES_REQ_CANNOT_EXECUTE_SERVICE

Restrictions:

Example:

```
int main (void)
{
  uint32_t error_code = SERVICES_REQ_SUCCESS;
  off_profile_t off_mode;

  error_code = SERVICES_get_off_cfg(services_handle,
                                    &runp,
                                    &service_error_code);

   g (error_code != SERVICES_REQ_SUCCESS)
  {
    sfuvso error_code;
  }
}
```

SERVICES_power_dcdc_voltage_control

Syntax:

Description:

Parameters:

Returns:
ErrorCode - SERVICES_REQ_SUCCESS, SERVICES_REQ_CANNOT_EXECUTE_SERVICE

Restrictions:

Example:

SERVICES_power_ldo_voltage_control

Syntax:

Description:

Parameters:

Returns:
ErrorCode - `SERVICES_REQ_SUCCESS, SERVICES_REQ_CANNOT_EXECUTE_SERVICE`

SERVICES_power_setting_configure

Syntax:

```
u  fefg fovn {
  POWER_SETTING_BOR_EN,
  POWER_SETTING_SCALED_CLK_FREQ,
} power_setting_t;
```

Description:

Parameters:

Returns:

SERVICES_power_setting_get

Syntax:

Description:

Parameters:

Returns:

SERVICES_power_stop_mode_raw_req

Syntax:

SERVICES_power_stop_mode_raw_req

Description:

Returns:

SERVICES_power_ewic_config_raw

Syntax:

SERVICES_power_ewic_config_raw

Description:

Parameters:

Returns:

SERVICES_power_wakeup_config_raw

Syntax:

SERVICES_power_wakeup_config_raw

Description:

Parameters:

Returns:

SERVICES_power_mem_retention_config_raw

Syntax:

SERVICES_power_mem_retention_config_raw

Description:

Parameters:

Returns:

SERVICES_power_m55_he_vtor_save_raw

Syntax:

SERVICES_power_m55_he_vtor_save_raw

Description:

Parameters:

Returns:

Clocks Services

Clock frequency definitions

RC
RC
XO
XO

SERVICES_clocks_select_osc_source

Syntax:

Description:

Type definitions:

Parameters:

Returns:

SERVICES_clocks_select_pll_source

Syntax:

Description:

Type definitions:

Parameters:

Returns:

SERVICES_clocks_enable_clock

Syntax:

Description:

Type definitions:

Parameters:

Returns:

SERVICES_clocks_set_ES0_frequency

Syntax:

Description:

Parameters:

Returns:

SERVICES_clocks_set_ES1_frequency

Syntax:

Description:

Parameters:

Returns:

SERVICES_clocks_select_a32_source

Syntax:

Description:

Type definitions:

Parameters:

Returns:

SERVICES_clocks_select_aclk_source

Syntax:

Description:

Type definitions:

Parameters:

Returns:

SERVICES_clocks_set_divider

Syntax:

Description:

Type definitions:

Parameters:

Returns:

SERVICES_clocks_get_clocks

Syntax:

Description:

Parameters:

Returns:

SERVICES_clocks_get_apb_frequency - OBSOLETE

Syntax:

Description:

Parameters:

Returns:

SERVICES_clocks_get_refclk_frequency - OBSOLETE

Syntax:

Description:

Parameters:

Returns:

SERVICES_clocks_setting_get

Syntax:

Description:

Parameters:

Returns:

SERVICES_pll_xtal_start

Syntax:

Description:

Parameters:

Returns:

SERVICES_pll_xtal_stop

Syntax:

Description:

Parameters:

Returns:

SERVICES_pll_xtal_is_started

Syntax:

Description:

Parameters:

Returns:

SERVICES_pll_clkpll_start

Syntax:

Description:

Parameters:

Returns:

SERVICES_pll_clkpll_stop

Syntax:

Description:

Parameters:

Returns:

SERVICES_pll_clkpll_is_locked

Syntax:

Description:

Parameters:

Returns:

SERVICES_pll_initialize

Syntax:

Description:

Parameters:

Returns:

SERVICES_pll_deinit

Syntax:

Description:

Parameters:

Returns:

Boot Services

```
HOST_CPU_0          A32_0
HOST_CPU_1          A32_1
EXTSYS_0            M55 HP CPU
EXTSYS_1            M55 HE CPU
```

SERVICES_boot_process_toc_entry

Syntax:

Description:

Parameters:

Returns:

Restrictions:

SERVICES_boot_cpu

Syntax:

Description:

Parameters:

Returns:

Restrictions:

SERVICES_boot_set_vtor

Syntax:

Description:

Parameters:

Returns:

Restrictions:

SERVICES_boot_reset_cpu

Syntax:

Description:

Parameters:

Returns:

Restrictions:

SERVICES_boot_release_cpu

Syntax:

Description:

Parameters:

Returns:

Restrictions:

SERVICES_boot_reset_soc

Syntax:

Description:

Parameters:

Returns:

Restrictions:

Crypto Services

SERVICES_cryptocell_get_rnd

Syntax:

Description:

Parameters:

None

Returns:

Restrictions:

Example:

```
int main (void)
{
  uint32_t ErrorCode = SERVICES_OK;
  uint64_t rnd_value;
  uint32_t service_error_code;

  ErrorCode = SERVICES_cryptocell_get_rnd(
              sizeof(uint64_t),/* random number/vector length in bytes*/
              &rnd_value,
```

```
            &service_error_code);
   g (ErrorCode != SERVICES_REQ_SUCCESS)
  {
    sfuvso ErrorCode;
  }
}
```

SERVICES_cryptocell_get_lcs

Syntax:

Description:

Parameters:

Returns:

Restrictions:

Example:

```
int main (void)
{
  uint32_t ErrorCode = SERVICES_OK;
  uint32_t lcs_state;
  uint32_t service_error_code

  ErrorCode = SERVICES_cryptocell_get_lcs(services_handle, &lcs_state,
&service_error_code);
   g (ErrorCode != SERVICES_REQ_SUCCESS)
  {
    sfuvso ErrorCode;
  }
}
```

These Services are not intended to be used directly by applications

SERVICES_cryptocell_mbedtls_hardware_poll

Syntax:

Description:

Parameters:

SERVICES_cryptocell_mbedtls_aes_init

Syntax:

Description:

SERVICES_cryptocell_mbedtls_aes_set_key

Syntax:

Description:

SERVICES_cryptocell_mbedtls_aes_crypt

Syntax:

Description:

SERVICES_cryptocell_mbedtls_sha_starts

Syntax:

Description:

SERVICES_cryptocell_mbedtls_sha_process

Syntax:

Description:

SERVICES_cryptocell_mbedtls_sha_update

Syntax:

Description:

SERVICES_cryptocell_mbedtls_sha_finish

Syntax:

Description:

SERVICES_cryptocell_mbedtls_ccm_gcm_set_key

Syntax:

Description:

SERVICES_cryptocell_mbedtls_ccm_gcm_crypt

Syntax:

Description:

SERVICES_cryptocell_mbedtls_chacha20_crypt

Syntax:

Description:

SERVICES_cryptocell_mbedtls_chachapoly_crypt

Syntax:

Description:

SERVICES_cryptocell_mbedtls_poly1305_crypt

Syntax:

Description:

SERVICES_cryptocell_mbedtls_cmac_init_setkey

Syntax:

Description:

SERVICES_cryptocell_mbedtls_cmac_update

Syntax:

Description:

SERVICES_cryptocell_mbedtls_cmac_finish

Syntax:

Description:

SERVICES_cryptocell_mbedtls_cmac_reset

Syntax:

Description:

SERVICES_update_stoc

Syntax:

Description:

Parameters:

Returns:

Restrictions:

Boot arguments

#define
#define
#define

typedef struct

-
-
-

SERVICES_Boot_Net_Proc

Syntax:

Description:

Parameters:

Returns:

Restrictions:

SERVICES_Shutdown_Net_Proc

Syntax:

Description:

Parameters:

Returns:

Restrictions:

Document History

Version	Change Log