Morse Micro

MM-IoT-SDK Software Release Notes

MCU Software Package

Table of Contents

1 Overview

The Morse Micro IoT SDK comprises the necessary host-side components for using an MCU host with an MM6108/MM8108 transceiver. This includes drivers, the upper MAC (UMAC) layer, the network stack, and RTOS.

The MM IoT SDK release package includes the following:

- **morselib:** a library containing the Morse Micro driver, UMAC, and WPA supplicant.
- **Source code:** The source code includes RTOS (FreeRTOS), network stack (LwIP, FreeRTOS+TCP, mbedTLS, coreMQTT, etc.), platform BSPs, and shim layers.
- **Example applications:** Various example applications to aid in understanding how to use the SDK and accelerate development efforts.
- **Additional libraries:** Developed by Morse Micro, including mmconfig for key-value persistent storage and mmagic for CLI interface generation.
- **API documentation:** for Makefile-based development.
- **Getting Started Guide:** for Makefile-based development.
- **Setup scripts:** development machine setup scripts and tools.

The SDK supports development on Linux, Windows, and Mac using PlatformIO. It also supports Makefile-based development on Linux. Ubuntu 22.04 x86_64 is the officially supported distribution.

2 Software Change History

2.1 Release 2.9.7

This section outlines significant changes from release 2.8.2.

2.1.1 New Features

- Soft AP functionality (beta - not for production)
    - Support for operation in AP mode with a limited feature set:
        - Maximum 4 connected STAs
        - Automatic and Dynamic channel selection is not supported
        - Soft AP example application only supports IPv4 and static IP addresses
        - See also the Soft AP subsection under Known Issues below
- Device Provisioning Protocol (DPP), push button mode functionality as STA (Beta)
    - DPP push button allows a device to be provisioned onto a network by pressing a button on both the STA and on the AP
    - Only supported on EKH05 platforms, as Public Key Accelerator (PKA) HW is required to meet timing requirements
- Support for 4-address frames in STA mode
    - Allows a STA to act as a layer 2 bridge
- Support for duty cycle burst mode
    - In regulatory domains subject to duty cycle limitations, burst mode allows a device to consume its allowance in a burst rather than spread evenly over the duty cycle period

2.1.3 Other Improvements

- Improved documentation of mmconfig parameters
- Added support for home channel dwelling during background scan
- Added MMWLAN API to configure listen interval
- Added support for registering a callback on station events
- Added support for appending user provided IEs to association IEs
- Updated mbedTLS to latest release v3.6.2
- Added API to query preserved failure logs
- Improved reliability of the WLAN chip health check system
- Added support for MCS8/9 rate control override for MM8108.
- Remove unsupported software scan
- Reduced power consumption of unused sensors on EKH05
- Added logging of return address to Hard Fault handlers, which can aid in identifying the instruction that triggered the fault.
- Improved EAPOL frame transmission reliability

- Enforced TWT configuration to occur pre-association
- Removed package files for unsupported MCU STM32-F429
- Extracted mmhal_flash API from mmhal.c
- Updated EKH05 CubeMX code generation to v6.14.1
- Added additional morselib.a builds to mm-iot-sdk-mm8108 so that it supports the same architectures as mm-iot-sdk.
- Added support for debug-only assertions using the macro MMOSAL_DEV_ASSERT()
- Removed support for MMAGIC CLI on EKH08-WB55 due to insufficient FLASH space
- Disabled LWIP address collision detection by default to improve DHCP link bring-up time
- Enabled DHCP by default on example applications and assumed target IP address is gateway address, where applicable (e.g., ping, iperf examples).
- Extended mmconfig configuration options for the AWS IoT example application
- Removed LED validation from porting assistant
- Updated ESP32 porting assistant to match main version
- Added support for debug builds in SDK Makefiles
- Added support for configuring the scan interval when running ACE script wlan-sta-connect.py
- Provided MM Chip MAC address to MMHAL read MAC address API
- Added scan dwell time configuration to mmconfig
- Added support for transceiver snooze in between scan attempts
- Added support for snooze before association to reduce power consumption
- Added support for disabling debugging in low power modes when using OpenOCD to improve power consumption profiling
- Removed unneeded SWO initialisation from main.c. OpenOCD initialises the SWO when required.
- Added PKA hardware acceleration support for EKH05 in mbedTLS
- Deprecated MMOSAL notification API
- Added a MMAGIC LLC synchronization mechanism to enable a controller to reattach to a running M2M agent
- Added MMAGIC CLI command to query WLAN link status
- Added TLS support in MMAGIC TCP core
- Added support for NTP in MMAGIC m2m
- Added support for a configurable timeout on MMAGIC m2m command responses
- Restructured autogenerated MMAGIC source code files to remove .def files and replace with .c files. In doing this several existing .c files were also renamed for clarity. This change should be transparent to users of the autogenerated makefile fragments.
- Added support for statically allocating MMPKT memory for m2m_agent demo application
- Reduced maximum mmagic streams to a sensible limit and allocated additional RAM to m2m_agent application to allow allocating the maximum streams.
- Added support for connecting to AP with higher operating bandwidth
- Reduced default scan dwell duration
- Added GB country code to regulatory domain database

- Added support for EU 2MHz channels and 100% Duty cycle
- Enabled subband transmission by default
- Added support for customer provided BCFs
- Added initial support for ESP32C6 platform
- Added a Kconfig for specifying the Morse chip firmware binary when building for ESP32
- Improved SDK documentation viewability for ESP-IDF version on Github
- Improved display of scan results in MMAGIC CLI
- Added support for clearing MMAGIC CLI variables
- Disabled OTA update support in aws_iot example application by default
- Reduced static RAM usage by 32KB when using MM8108 devices
- Added ESP32C3 support to mm-iot-sdk-esp32

2.1.4 Bug Fixes

- Fixed `mmhal_get_time()` API missing explicit void argument
- Fixed reporting TWT Requester S1G capabilities as "Not Supported"
- Ensure MAC address persists over the app lifetime
- Fixed RAW priority 0 being treated as disabled
- Reduced power consumption on MM8108-EKH05 when HaLow chip is off
- Improved robustness of porting assistant BUSY pin testing
- Fixed MMAGIC CLI unable to set full IPV6 address
- Fixed memory leak on failed TX in MMAGIC Agent LLC.
- Fixed potential memory leak in m2m_agent application for SPI datalink buffer TX.
- Fixed MMAGIC CLI becoming unresponsive after executing one-shot system deep sleep command `sys-deep_sleep one_shot`
- Fixed m2m_agent SPI data link recovery from receive error.
- Fixed ignored MMAGIC wlan core configuration parameters
- Fixed MMAGIC CLI maximum wlan password length not matching MMWLAN maximum
- Improved MM8108 TCP throughput stability in environments experiencing packet loss

2.1.5 Known Issues

2.1.5.1 Makefile Infrastructure

- Building fails when MM IoT SDK package directory has a space in the path
    - The MM IoT SDK package must be unpacked into a directory that does not include a space in its name.

2.1.5.2 Stability

- Timeouts with FreeRTOS time slicing disabled
    - FreeRTOS time slicing is enabled by default in the MM-IOT-SDK FreeRTOS configuration. If this is disabled, timeouts may be observed when sending commands to the MM chip.
- System time drifts when host deep sleep is enabled
    - System time may drift from wall clock time when deep sleep is enabled. This may result, for example, in iperf taking longer than expected to complete. Drift of 1-2% has been observed on STM32U575 and 3-4% on STM32WB55, but this may vary with level of activity.
- DPP must be stopped explicitly before shutdown
    - If DPP has been started (via `mmwlan_dpp_start()`) it must be stopped explicitly via `mmwlan_dpp_stop()` prior to invocation of `mmwlan_shutdown()`.
- Assert after consecutive RNG failures
    - On STM32 platforms, in cases where multiple threads access the hardware random number generator (RNG), a priority inversion may occur resulting in an assertion in `mmhal_random_u32()`. This can be mitigated by protecting access to the STM32 RNG API with a mutex.

2.1.5.3 Throughput

- Low throughput with open security due to action frames being dropped
    - When connected with open security (no encryption), action frames are dropped on receive. The primary impact of this is that block ack negotiation fails, meaning A-MPDU aggregation is not enabled, and thus resulting in lower throughput. This does not affect SAE or OWE security modes.

2.1.5.4 Example Applications

- Failures when lwIP built with LWIP_IPV4=0
  - If IPv4 is disabled when LWIP is built (i.e., IPv6 only) then there may be issues with the IP stack not working as expected. The workaround is to always enable IPv4.
- CLI application requires country code to be programmed to config store
  - The CLI application will assert during initialization if a valid country code has not been programmed to the config store. Instructions for programming config store can be found in the Getting Started Guide.
- Bootloader/OTAU support
  - The example bootloader and over-the-air-update (OTAU) support in the aws_iot example has not been tested to a level where it is recommended for production use. The example bootloader may be deprecated in a future release.

2.1.5.5 Standby/Offload Features

- The Standby and Offload features of the MMWLAN API are in development and may not be fully functional.
  - Further improvements are planned in future releases that may result in minor API changes.

2.1.5.6 Soft AP

- Power save is not supported on connected STAs
  - Soft AP does not support buffering of traffic for power saving STAs. Therefore, 802.11 power save must be disabled on STAs connecting to the AP.
- Traffic is not forwarded between STAs
  - Soft AP does not forward frames between connected STAs. Therefore it is not possible for connected STAs to communicate with each other.
- Soft AP does not correctly transmit group addressed frames
  - Group addressed frames are transmitted unencrypted, even when encryption is enabled, meaning that they are not able to be decoded by receiving STAs.
- Recovery from health failures is not supported in Soft AP mode
  - If a health check failure occurs while operating in Soft AP mode, it will not return to an operable state (for example, beacon transmission will not resume).

2.1.6 Migration from 2.8.2

The following is a list of significant changes from the previous release of the MM-IoT-SDK. This is not an exhaustive list and does not include new feature additions or changes to example applications.

2.1.6.1 mbedTLS configuration

For mbedTLS configuration files that use `mmhal_get_time()` for `MBEDTLS_PLATFORM_TIME_MACRO`, the definition must be updated to the following to avoid compilation errors:

```
#define MBEDTLS_PLATFORM_TIME_MACRO(x)          mmhal_get_time()
```

2.1.6.2 MMOSAL API

Newly added functions:

- `mmosal_printf()`: OS abstracted version of printf used by morselib
- `mmosal_extract_failure_info()`: Extract the oldest un-viewed failure log entry

The task notification API (`mmosal_task_wait_for_notification()`, `mmosal_task_notify()`, `mmosal_task_notify_from_isr()`) has been deprecated. This API is no longer used by morselib and it may be removed from the MMOSAL API in future.

The macro `MMOSAL_DEPRECATED_API_ENABLED` has been added to `mmosal.h`. If set to `0` then deprecated MMOSAL API will be excluded by the preprocessor. The default value is `1` to allow for backward compatibility, but it can be overridden to 0 for increased strictness.

2.1.6.3 MMWLAN API

Newly added API:

- `morse_chip_id_string` field added to `mmwlan_version` struct.
- `home_channel_dwell_time_ms` field added to `mmwlan_scan_config` struct.
- Duty cycle API (`mmwlan_set_duty_cycle_mode()` and `mmwlan_get_duty_cycle_stats()`) added.
- STA event callback API added (`sta_evt_cb` and `sta_evt_cb_arg` fields added to `mmwlan_sta_args` struct). *(Beta)*
- `extra_assoc_ies` and `extra_assoc_ies_len` fields added to `mmwlan_sta_args` struct.
- `use_4addr` field added `mmwlan_sta_args` struct.
- DPP API added (`mmwlan_dpp_start()` and `mmwlan_dpp_stop()`). *(Beta)*
- `dwell_on_home_ms` field added to `mmwlan_scan_args` struct.
- Soft AP API added (`mmwlan_softap_enable()` and `mmwlan_softap_disable()`). *(Beta)*

API changes:

- `MMWLAN_SCAN_DEFAULT_DWELL_TIME_MS` reduced from 105 to 30 due to improvements in scan implementation.

2.1.6.4 MMHAL API

The behavior regarding invocation of the function `mmhal_read_mac_addr()` has changed. This function will now only be invoked once after boot of the transceiver (e.g., via `mmwlan_boot()`) and the value will be cached, where previously it may have been invoked multiple times.

2.1.6.5 MMAGIC

Restructured auto-generated MMAGIC source code files to remove .def files and replace with .c files. In doing this several existing .c files were also renamed for clarity. This change should be transparent to users of the auto-generated makefile fragments.

Low-level MMAGIC Controller API added: `mmagic_controller_agent_sync()` and `mmagic_controller_request_agent_reset()`.

New API: *TLS*, *MQTT*, and *NTP*. Currently supported in M2M mode only.

Due to FLASH constraints, support for the MMAGIC CLI has been removed on the mm-ekh08-wb55 platform.

2.1.6.6 MQTT agent

MQTT agent has been moved out of the `aws-common` component and into its own `freertos-libs` component. This enables using the MQTT agent without introducing a dependency on AWS, such as in the MMAGIC MQTT module.

Existing Makefile-based applications using MQTT via AWS will need to include an additional Makefile fragment in their root Makefile::

```
include $(MMIOT_ROOT)/mk/mqtt-agent-task.mk
```

This change should be transparent to users of the auto-generated makefile fragments.

2.1.6.7 Makefile infrastructure

Setting of Makefile CFLAGS for optimization has been moved from misc.mk to Makefile for each example. Projects relying on this flag being set in misc.mk will need to be updated to set the flag in their Makefile.

2.1.6.8 Morse Firmware/BCFs

⚠ The firmware version has been updated from 1.15.3 to 1.16.2. Devices may need to have their BCF files updated to the latest version if they want the latest features on MM6108/MM8108.

3 System Details

3.1 Components

Component	Version
Morse firmware	1.16.4
Regulatory Database	2.4.1
FreeRTOS kernel	10.5.1
LWIP	2.2.0
FreeRTOS + TCP	4.3.1
mbedTLS	3.6.2
STM32U5 BSP	1.4.0
STM32WB55 BSP	1.14.1
BCF API	12.1.0 (backwards compatible to 8.0.0, but features may be limited)

3.2 System Memory Requirements

Supported architectures:

- ARM Cortex-M4F
- ARM Cortex-M7F
- ARM Cortex-M33F
- ESP32 (beta)

Memory	Recommended
RAM	256 KB+
Flash	2 MB+

Space required will depend on the specific application. This may not be sufficient to support, for example, cloud connectivity, where application-layer protocols such as TLS and MQTT are required.

3.3 HaLow Throughput Performance

Throughput is measured at 8 MHz, with RTS disabled (unless otherwise stated), SGI enabled, power-save enabled, using Lightweight IP (LWIP), and with iperf run for 60 seconds, taking the median of 5 iterations.

AP is running version 1.16.4 drivers/firmware.

Test Type	Platform	UDP (Mbps)		TCP (Mbps)	
		STA -> AP	AP -> STA	STA -> AP	AP -> STA
Single STA	MM6108 EKH08-WB55	5	9	4	5
	MM6108 EKH05 SDIO	22	20	7	9
	MM6108 EKH05 SPI	17	17	6	8
	MM8108 EKH05 SDIO	23	29	8	10
	MM8108 EKH05 SPI	19	21	7	9
Multi STA (4 STA cumulative)	MM6108 EKH08-U575 SPI	19	13	9	13
	MM8108 EKH05 SPI	24	24	10	15

4 Revision History

Release Number	Release Date	Release Notes
Version 1	1 Oct 2025	Initial release

Morse Micro

Morse Micro Pty. Ltd. Corporate Headquarters

Level 8, 10-14 Waterloo Street, Surry Hills, NSW 2010, AUSTRALIA

Email: sales@morsemicro.com