

Release Notes 2.5.0

Table of Contents

- [Table of Contents](#)
- [Updates and new features](#)
 - [Features/Quality](#)
 - [mira_net_rejoin\(\)](#)
 - [Frontend control on SoftDevice builds use 'idle' as default state](#)
 - [Mira version API](#)
 - [Improvements](#)
 - [ABI change](#)
 - [rf_slots API changes](#)
 - [Renaming of ISR routines for MKW41z](#)
 - [Delivery of startup file for MKW41z](#)
 - [Bugfixes](#)
 - [Incorrect return types in mira_mem API](#)
 - [Border Gateway unicast messages may be sent as broadcast](#)
 - [Border Gateway crash at boot](#)
 - [Frontend configuration via license not possible on MiraOS](#)
- [Known issues](#)
- [Limitations](#)
- [Upcoming changes](#)
 - [Deprecation of nRF based builds without Nordic Semiconductor SoftDevice](#)

Updates and new features

Features/Quality

`mira_net_rejoin()`

A function to leave the current network and immediately start joining a network with the same credentials. This is useful when minimising network rebuild times by doing controlled restarts of an entire network, rather than just the root, during maintenance or setup. Previously, nodes would have to be fully rebooted to achieve the same result. Now, the application can call `mira_net_rejoin()`, and the nodes can be kept running.

Frontend control on SoftDevice builds use 'idle' as default state

Previously Mira used 'bypass TX' as default FrontEnd Module (FEM) state on SoftDevice builds. This enabled concurrent BLE operation as the FEM blocks RF signals in the 'idle' state. Leaving the FEM in 'bypass TX' does, however, increase current consumption (above the 200uA range for a meshing node). Mira now uses 'idle' as default state on all platforms. Along with this change, we have also changed the MWA_N2 FEM configuration in such a way that the 'idle' state has the same pin configuration as the 'bypass TX' state. This means the following:

Users of our MWA_N2 radio module

- No change in behaviour, given that our FEM configuration is used. Concurrent BLE is possible, but at a high current consumption.
- To make an MWA consume less current, at the cost of concurrent BLE, a custom frontend configuration will have to be made. There are examples of this. See the [frontend documentation](#).

Users of custom FEMs

- To maintain the same behaviour as previous releases the FEM configuration may have to be updated. See the [frontend documentation](#).
- For this reason, configuration via the license file is discouraged in the scenario, to avoid problems when upgrading.

Mira version API

A new API function has been added to provide a string with the running Mira version. See `mira_version.h`.

Improvements

ABI change

The ARM VFP ABI (hard floating point) ABI is now used instead of ARM EABI ABI for the nRF chipsets. This is the same ABI used in versions before 2.4.0.

rf_slots API changes

A new mandatory cleanup section in the rf-slots process handler is added: RF_SLOTS_BEGIN_CLEANUP(). Code in this section is automatically executed when a slot is either finished or aborted. See rf-slots based BLE beacon example for more information.

The struct rf_slots_radio_config_t has changed format to allow for more control over radio format. See the [API documentation](#) for more details.

Keep in mind that 802.15.4 modulation and prioritization between radio slots, which enables coexistence between slow rate Mira and arbitrary protocols, is still not available in the rf_slots API.

Renaming of ISR routines for MKW41z

To avoid naming conflicts, all ISR routines in Mira have been renamed with a mira_* prefix. Users relying on ISR routine names within Mira need to update accordingly.

Delivery of startup file for MKW41z

The file "startup_MKW41Z4.S" is now compiled by the user, instead of being compiled and linked in the library. This allows users to more easily make modification to the file if necessary.

Bugfixes

Incorrect return types in mira_mem API

Functions mira_mem_set_alloc_callback() and mira_mem_set_alloc_buffer() now returns the type mira_status_t. Previously, while returning the correct value, they incorrectly returned the type mira_size_t.

Border Gateway unicast messages may be sent as broadcast

Resolved an issue where unicast messages could be sent as broadcast, resulting in many unexpected replies. This bug was limited to the Border Gateway.

Border Gateway crash at boot

Resolved an issue where the Border Gateway would crash if it received a message during its boot sequence.

Frontend configuration via license not possible on MiraOS

The ability to configure a frontend via the license file was not supported in MiraOS, contrary to what was mentioned in the documentation. The feature was only supported on MiraMesh and Border Gateway platforms. Mira now supports frontend configuration via the license file and configuration areas on MiraOS and Border Gateway. Frontend configuration via configuration areas is unsupported on MiraMesh, due to configuration areas not being supported on that platform. For more information, please refer to the [configuration area documentation](#).

Known issues

See previous release notes.

Limitations

See previous release notes.

Upcoming changes

Deprecation of nRF based builds without Nordic Semiconductor SoftDevice

Builds on the nRF52840 and nRF52832 will require the Nordic Semiconductor SoftDevice in version 2.6.0. Builds without the SoftDevice will be removed. Contact LumenRadio if your product can't be changed to use the SoftDevice.

