**Neptune Wireless Sensor Module**

**Embedded Software Release Notes**

**Release Date: 25 June 2024**

**RadioFaas Part Numbers:**

601010-000001 – R01 – 2.0.0.0001 [Wireless Sensor Module Application]

**Features and Improvements:**

* Addition of Telit Wifi new commands

**Wireless Sensor Module Programming Instructions (manual):**

1. Connect the JTAGICE mkII debugger and launch Atmel Studio 6.1 (Can support above version it will automatically generate new version. atsln .
2. Power on the Wireless Sensor Module.
3. Click 'Tools->AVR Programming' and select the appropriate JTAG/ISP connection.
4. Select the ‘Production file’ tab.
5. Browse to the ELF image file ‘**SensorModuleWirelessBoot\_6\_1**.elf’.
6. Check the boxes to select the following sectors: ‘flash’, ‘eeprom’, ‘fuses’, and ‘lock bits’.
7. Make sure that the ‘erase before programming’ and ‘verify after programming’ boxes are checked.
8. Click the ‘Program’ button and verify that the image and verification complete successfully.
9. Select the 'Memories' tab.
10. In the Flash Memory section, select the application image file ‘**SensorModuleApp64M1\_6\_1.hex'** and click the ‘Program’ button (make sure the 'Verify flash after programming' option is selected and that the ‘Erase device before programming’ option is de-selected).
11. Verify that the application image programming and verification completes successfully.
12. Disconnect power.
13. Power on board for 5-7 seconds to update the firmware on Telit module. (only first time required if this step is done once it directly boots up with the new firmware)
14. Power off the board
15. Your board is ready now.

**FUSE SETTINGS**

The fuse bits are programmed automatically as part of the bootloader ELF image, as described above. These bits can be verified in the Atmel Studio 6 'Tools->AVR Programming->Fuses' menu with the emulator connected via JTAG/ISP.

The following values should be present:

Extended: 0xFE

High: 0xD8

Low: 0xFF

**SERIAL NUMBER PROGRAMMING**

After the Sensor Module firmware has been programmed, the device serial number can be programmed via the device’s CAN interface. Sending a message with CAN ID of 0x24 with an 8-byte payload containing the ASCII serial number string (typically of format ‘LBW-xxxx’) will force the number to be written to EEPROM.

**HARDWARE VERSION PROGRAMMING**

The device hardware version can be programmed via the device’s CAN interface. Sending a message with CAN ID of 0x25 with a 5-byte payload containing the ASCII version string (typically of format ‘Rxx’) will force the string to be written to EEPROM. Unused bytes in the payload may be left as spaces or null characters.

**HARDWARE PART NUMBER PROGRAMMING**

The device hardware version can be programmed via the device’s CAN interface. Sending a pair of CAN messages is necessary to specify the full part number string (typically of format ‘150505-xxxxxx’). The first segment must be transmitted using CAN ID of 0x2E with payload of 8 ASCII characters (‘150505-x’). The second segment must be transmitted using CAN ID of 0x2F with payload of 5 ASCII characters (‘xxxxx’). When received, these messages will force the string segments to be written to EEPROM.