Library Assignment for RN2483

A popular library for using RN2483[1][2] in the Arduino framework has been published by The Things Network [3] (TTN-library). This library is compatible with the HAN-IoT-node [4] and used in the HAN-KISS-Node [5].

For the purpose of education, a "derivative" of the TTN-library was created to support the use of other devEUI than the one in the RN2483. This derivative is found in the HAN-IoT-Node as header [6]- and implementation [7]-file.

One serious downside of the TTN-library is code-size of this library. The size hinders the use of multiple and complex sensors on the ATmega32U4 that is used on the The Things Uno (that is used in the HAN-IoT-node) and the KISS-LoRa.

A library for the RN2483 shall be developed to be used in the HAN-KISS-LoRa and Han-IoT-node. This will enable the addition of more and or complex nodes to these devices.

[1] https://ww1.microchip.com/downloads/en/DeviceDoc/50002346C.pdf
[2] https://ww1.microchip.com/downloads/en/DeviceDoc/40001784B.pdf
[3] https://github.com/TheThingsNetwork/arduino-device-lib
[4] https://gitlab.com/wlgrw/han-iot-node/
[5] https://gitlab.com/wlgrw/han-iot-kiss-lora
[6] https://gitlab.com/wlgrw/han-iot-node/-/blob/master/Software/Arduino/HAN_IoTShield_HelloWorld.ino/TheThingsNetwork_HANIoT.h?ref_type=heads
[7] https://gitlab.com/wlgrw/han-iot-node/-/blob/master/Software/Arduino/HAN_IoTShield_HelloWorld.ino/TheThingsNetwork_HANIoT.cpp?ref_type=heads

Assignment 1

The objective of this assignment is to minimize code footprint while maintaining compatibility with the TTN-library.

The assignment is:

> Refactor the TTN-library to reduce the code footprint by removing all unnecessary features.

The following additional requirements apply:

- The new library shall preserve full LoRaWAN compliancy without any overlap to the implementation in the RN2483.
- Compatibility with the TTN-library shall be preserved. The following public functions shall be implemented:

Tabel 1: Public functions for new class.

```
<<className>>(Stream &modemStream, Stream &debugStream, ttn_fp_t fp,
    uint8_t sf = TTN_DEFAULT_SF, uint8_t fsb = TTN_DEFAULT_FSB);
void reset(bool adr = true);
void resetHard(uint8_t resetPin);
void showStatus();
size_t getHardwareEui(char *buffer, size_t size);
size_t getAppEui(char *buffer, size_t size);
uint16_t getVDD();
void onMessage(void (*cb)(const uint8_t *payload, size_t size, port_t
    port));
bool provision(const char *appEui, const char *appKey);
bool provision(const char *devEui, const char *appEui, const char
    *appKey);
bool join(const char *appEui, const char *appKey, int8_t retries = -1,
    uint32_t retryDelay = 10000);
bool join(const char *devEui, const char *appEui, const char *appKey,
    int8_t retries = -1, uint32_t retryDelay = 10000);
bool join(int8_t retries = -1, uint32_t retryDelay = 10000);
bool personalize(const char *devAddr, const char *nwkSKey, const char
    *appSKey);
bool personalize();
ttn_response_t sendBytes(const uint8_t *payload, size_t length, port_t
    port = 1, bool confirm = false, uint8_t sf = 0);
ttn_response_t poll(port_t port = 1, bool confirm = false);
void sleep(uint32_t mseconds);
void wake();
void saveState();
void linkCheck(uint16_t seconds);
uint8_t getLinkCheckGateways();
uint8_t getLinkCheckMargin()
```

Assignment 2

To distribute your new library and aid the adoption, an industry standard shall be applied.

The assignment is:

Publish the new library as an Arduino Library.

The following additional requirements apply:

- The library:
  - Meets the Arduino Library specification [8][9].
  - Has documentation on how to use the library in a sketch.
  - Has Example sketches to be used in te Arduino Framework (Hello world) for the various configurations:
    - OTAA with devEUI of the RN2483
    - OTAA with devEUI specified by the user.
    - Using the payload encoder from another team.

Assignment 3

Transferring and sharing knowledge is an important task of an engineer. This also applies to the work of this assignment because the goal is to have the library being developed to be used for many years and adopted by the community.

The assignment is:

Present your work as part of a technical meeting.

The following additional requirements apply:

- Demonstrate the correct operation of your work.
- Present the important engineering decisions that resulted in the implementation that is presented.
- Expect questions during the technical meeting about your implementation.

[8] https://arduino.github.io/arduino-cli/0.19/library-specification/

[9] https://docs.arduino.cc/hacking/software/Libraries

Architecture principles

The following Architecture principles [10] are to be respected:

- Programming language and standard: The library shall be code using C++11 standard; Use of a recent standard enables cleaner and more concise code.
- Version management: The libraries shall be published in a public library using git; The libraries shall be reviewable and maintainable by the open-source community to enable usage and contributions; The library is accessible to anyone for peer review and the code (or parts of it) can be used for reuse.
- Licence: The libraries shall be published under a Commons Creative licence CC BY 4.0 Deed Attribution 4.0 International or equivalent; The work is not intended for commercial use but for educational purposes and shall be made available to others. This will enable peer review as well as use, adaptations, and contribution to the code. At the same time the work of the authors shall be honoured and respected; the code will be available to other teams too.
- Testing: The libraries shall be tested using unit tests; Software can only be delivered when tests prove that the required operation is verified. For testing, both a test framework as well as tailored code for the purpose of (unit)testing, may be used. The test-code is part of the software, the test-results are part of the documentation; This approach aligns with the best practices in software development by prioritizing quality, reliability, and maintainability through a structured and thorough testing process.
- Documentation; All software shall be documented using Doxygen [11]; Documenting software is crucial for its long-term success, ensuring maintainability, usability, and scalability while facilitating efficient collaboration among developers and stakeholders; Documentation requires extra time to work on.
- <Name>, <Statement>, <rationale>, <implications>

[10] https://pubs.opengroup.org/architecture/togaf8-doc/arch/chap29.html

[11] https://www.doxygen.nl/