Firmware (LoRaWAN Modbus Gateway)

MOVED

https://docs.lobaro.com/xwiki/bin/view/Main/LoRaWAN%20Modbus%20Gateway/Firmware%20%28LoRaWAN%20Modbus%20Gateway%29/

Typ	Version	File	Installation
Debian Package	1.9.0	lobaro-modbus-server_1.9.0_linux_armv6.deb	`wget https://s3.eu-central-1.amazonaws.com/public.lobaro.com/releases/lobaro-modbus-server/lobaro-modbus-server_1.9.0_linux_armv6.deb` `sudo dpkg -i lobaro-modbus-server_1.9.0_linux_armv6.deb`

Installation

```
wget <firmware-url>
sudo dpkg -i <file-name>
```

Without internet access use SCP (e.g. WinSCP for Windows) instead of wget to copy the file onto the gateway.

Changelog

v1.9.0 - 2023-12-11

Added

- Parse objectJSON string to JSON object, allowing usage of items in it. Thus, the modbus server can be used with the "new" chirpstack JSON marshaller

Fixed

- Keep restarting modbus server by increasing the time between restart attempts, and not trigger the start limit watchdog

v1.8.2 - 2023-12-01

Changed

- Restart modbus server if connection to serial port is lost, instead of terminating the service.
- Ensure systemd always tries to restart the server (unless stopped by systemctl), but delay successive restarts up to 10s.

Fixed

- Return errcode (1) if the server exits due to an error in a service (e.g. Modbus service loses connection to its serial port).

v1.8.1 - 2023-11-13

Table of Contents

Added

- Method for safely setting and getting LoRaWAN activations map, via its mutex
- Tests for LoRaWAN activations: JSON export, setting, getting

Changed

- Set and Get LoRaWAN activations map via its method, to avoid concurrent writes or iterations
    - Was possible when Store was saved while Activations were updated

v1.8.0 - 2023-05-24

Added

- Variable downlinks via Modbus
    - Set trigger register and (max) downlink length Len in modbus registers (16 bits/2 bytes each)
    - Write downlink content to registers behind trigger register (max Len registers)
    - Write `<port><length>` in hex to trigger register to send downlink
        - E.g. write `8009`(hex) to trigger register to send 4 + first half of 5th register following it (=9 bytes) as a downlink to port 128 of the device

Changed

- Set downlink (trigger) register to `0` instead of `1`
- Use modbus slaveID from config file
- Default modbus slaveID to `1`
- Upgrade go dependencies

v1.7.0 - 2023-05-11

Added

- Chirpstack API calls via gRPC for future use

Changed

- Update go dependencies
    - Update jwt usage, fix token parser test

Deprecated

- Chirpstack service Profile no longer used for gateway creation

v1.6.1 - 2023-05-04

Changed

- No storage of downlink registers in `register-map.json`
    - Avoids issues with former downlink registers not losing their state after a config change

Fixed

- Bug where device uplinks would be missed because of MQTT legacy topic subscription

v1.6.0 2023-04-17

Added

- Add local gateway via REST to Chirpstack, determining gateway EUI from packet forwarder config.

v1.5.0 - 2023-04-04

Added

- Downlinks triggered by modbus register writes
    - Value in register will be written value if last downlink was successful, 0xFFFF if last downlink failed

Fixed

- Bug where 32-bit values wouldn't be stored properly in modbus registers

v1.4.1 - 2022-09-01

Fixed

- Bug where devices are not created in Chirpstack

v1.4.0 - 2022-08-31

Added

- Gateway Userinterface on port 8081

v1.3.1 - 2022-06-28

Fixed

- Handling of verbose flag
- verbose logging for config loading

v1.3.0 - 2021-02-01

Added

- Device activations are saved and restored

Changed

- Setup sets 777 access rights on config file and dir

1.2.1

Added

- devName config parameter to specify an optional device name

Fixed

- modbus "disabled" config parameter was always true

1.2.0

Added

- Allow to configure modbus serial connection parameters
    - modbus.dataBits: 8
    - modbus.parity: "even" # no, even (default), odd
    - modbus.stopBits: 1 # 1 (default), 1.5, 2

1.1.1

Added

- Support int32, uint32, float32 register types

Fixed

- Bug where register data is not saved

1.0.7

Fixed

- Allow async access on modbus registers. Might solve invalid value read-out.
- Persist data only once per minute
- Persist data on exit

1.0.6

Fixed

- Reconnect to MQTT on disconnect
- possible endless loop on MQTT receive

1.0.3

Fixed

- modbus register readout (was always 0)