comtac AG
Bettschen Elektronik
Allenwindenstrasse 1
CH-8247 Flurlingen
info@comtac.ch
www.comtac.ch
T +41 52 647 30 30

LPN Modbus easy SW / Specification V0.19

Comtac AG
CH-8247 Flurlingen

History

Date	Description	issue of document	Firmware version
2017-03-27-Kd	First Release REV00 V00.00		
2017-05-02-Kd	V0.01 update of USB-CDC function, Network choice Public/Private per CFG.TXT		
2017-11-30-Kd	V0.02 Register address clearly documented		
2017-12-18-Zs	V0.03: Formating of the document changed		
2018-01-18-Kd	V0.06: Implemented SendOnChange and the Port 4 (Transceiver)		
2018-01-23-Kd	V0.08: Implemented RTU (2*Interdelay)		
2018-04-11-Kd	V0.10: Implemented FrequencyPlan (corr. ConfirmTx)		
2019-06-04-Zs	V0.14: downlink Example implemented		
2019-11-29-Kd	V0.17: Baudrate limits 600..115200 (old Version could got Resets with > 57600Baud)		
2020-04-29-Kd	V0.18: FC1 + FC2 were reversed		
2020-06-09-Kd	Snr comment wrong		
2020-11-24-Kd	REVxx V00.19: New Ports 101 LoRa-Config., Port 105 Reset and new configuration parameters ConfirmedTries, DownlinkWatchdog and ResetInterval. **RndTime is automatically changed to 1 after the update, if old App. Version <0.19 and RndTime was 10. With RndTime 1, no sendinterval restart after the first uplink will be done.**	E1332-LPN_ModbusEasy_Bridge_EN_V0.19	00.19
2021-03-22_Su	Decription of Parameters: -ConfirmedTries -DownlinkWatchdog -ResetInterval. added.	E1332-LPN_ModbusEasy_Bridge_EN_V0.19-1	00.19

Changes are added in this history, if a new version has been issued.

Contents

1 Features

The LPN Modbus Bridge acts as a Modbus Master (Client) on the RS-485 bus. When configured, the bridge reads Modbus Registers and sends them via LoRaWAN as a Class C device.

1.1 Function DIP switch (only implemented on REV00 hardware)

SW-1 is evaluated at each uplink and SW-2 + SW-3 are evaluated only when powered on.

DIP-switch No. [0...X]	Function/Meaning	Remarks
1	Default off	LoRaWAN TxConfirmed uplinks OFF or ON
2	Default off	LoRaWAN device activation OFF= APB (ActivationByPersonalization); ON=OTA (OverTheAir)
3	Default off	Network type OFF = Public(Preamble = 0x34); ON = private (preamble = 0x12)

1.2 Function of the Buttons

Button	Function/Meaning	Remarks
SEND	On Power Up	When only SEND button is held while switching on, the boot loader is activated (red LED flashes briefly on and all other LED lights).
SEND	During Startup	After power-up, the user got 2 seconds time to perform a special function, which will be indicated by alternately flashing orange and red (100ms clock) LED. If SEND button is pressed, the USB will be in USB-CDC Mode (Virtual COM Port), used for special configuration. A special function is acknowledged by a fast flashing of the green LED for 1 second.
SEND	During operation	A Confirm-Uplink (port 0 if no other uplinks are pending) is sent by pressing the SEND button. If a connection has not yet been established with OTA, a JoinRequest is sent before.
CHECK	During Startup	After power-up, the user got 2 seconds time to perform a special function, which will be indicated by alternately flashing orange and red (100ms clock) LED. If CHECK button is pressed, LoRa TimeOnAir (minimum pause times between the sending) is ignored. A special function is acknowledged by a fast flashing of the green LED for 1 second.
CHECK	During operation	CHECK Button triggers all Modbus measurement intervals. Pressing the button for more than 3s will trigger a software reset. When you start up, the orange and red LEDs will flash simultaneously (100ms ON 100ms OFF) until the CHECK button is released again.

Reset of the device configuration during startup:
If both buttons are pressed, the LoRa configuration (CFG.TXT) is reset to the default values.

1.3 Function of LEDs

Blink variants of the LEDs: 12%-> 0.7s off + 0.1s on; 50%--> 0.4 s off + 0.4 s on; 88%-> 0.1 s off + 0.7 s on

After switching on, all LEDs light up for 0.5 seconds, if the LEDs remain lit and the red LED flashes briefly, the bootloader is active.

LED	Function	Remarks
Red	Displays Error	Off: In order. 12%: No Modbus registers configured.
Orange	Modbus Status	Off: Not initialized. 12%: No echoes on the RS485 bus -> Check bus (short circuit or missing bus termination) 50%: No response from the addressed Modbus device (evaluate Modbus error). 88%: Modbus command responds with one exception (evaluate Modbus exception). On: In order.
Green	Power supply	Lights up when power is available. During startup, a special function selected by the buttons is confirmed by a fast flashing (100ms ON 100ms OFF). During operation, a short extinguishing (100ms) of the LED indicates a LoRa data reception (downlink from the server).
Blue	LoRa Status	Off: Not initialized. 12%: Wait for OTA-Joining or wait until the start-up window has expired. 50%: No server downlink received (only for confirmed uplinks or SEND button). 88%: Uplink in progress or wait for LoRa-TimeOnAir enable (check data rate). On: In order (currently no uplinks to send).

2 LoRa Up- and Downlink

Commands from the server to the Node (LoRa Bridge) are downlinks and from the node to the server are uplinks. In the LoRaWAN, all uplinks are provided with a CRC by default, but the downlinks are not.

2.1 LoRa uplink payload structure on Port3

Data is sent in Big Endian format (MSB first).

Confirmed downlinks, with at least one payload byte, are answered with an uplink.

These data are also sent in the send interval. **The payload length is 2..34 bytes, depending on the configured Modbus registers.**

Byte No. [0...X]	Function/Meaning	Remarks
0	Status Modbus REG00..07	REG00..07 Status mask 0=Error or not used 1=Data Ok (MSBit=REG00 .. LSBit=REG07)
1	Status Modbus REG08..15	REG08..15 Status masks 0=Error or not used 1=Data Ok (MSBit=REG08 .. LSBit=REG15)
2 + (2*n)	REGn	REGn register value MSB (last valid value) (optional)
3 + (2*n)	REGn	REGn register value LSB (last valid value) (optional)
…	REGn	Moore registers of the Modbus device if configured

After the last configured register, no more data is sent, so the note (optional).

2.1.1 LoRa uplink payload example

Here one Modbus register is configured to readout periodically, below the payload:

```
80 00 00de
```

Byte No. [0...X]	Function/Meaning	Remarks
0	Status Modbus REG00..07	**0x80** = 0b1000 0000 = REG00 Data Ok REG00..07 Status mask 0=Error or not used 1=Data Ok (MSBit=REG00 .. LSBit=REG07)
1	Status Modbus REG08..15	**0x00** = Error or here not used REG08..15 Status masks 0=Error or not used 1=Data Ok (MSBit=REG08 .. LSBit=REG15)
2 + (2*n) 3 + (2*n)	REGn	**0x00 de** = 222 is the value which is read from the Modbus register configured

2.2 LoRa downlink payload structure on Port4 (Transceiver Port)

Confirmed downlink will trigger an uplink answer.

Byte No. [0...X]	Function/Meaning	Remarks
0	DevAddr	Modbus device address
1	FC	Functioncode
…	Data	See http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

Example message:
0x02 0x06 0x00 0x00 0x00 0x01

Byte No. [0...X]	Function/Meaning	Remarks
0	DevAddr	0x02 -> Modbus Adress
1	0x06	Functioncode
2/3	0x00 0x00	Modbus Register 0x01 – 0x01 = 0
4/5	0x00 0x01	Value 0x01

2.3 LoRa uplink payload structure on Port4 (Transceiver Port)

Answer to a confirmed downlink on Port 4.

Byte No. [0...X]	Function/Meaning	Remarks
0	Error code	Answer state
1	DevAddr	Modbus device address
2	FC	Functioncode
…	Data	See http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

Error codes:

0: No Error
1: Not initialized
2: In use (Busy)
3: No echo received -> Check RS485 Bus shorten
4: Function code not supported
5: No answer in the given time
6: Interframe timeout
7: UART parity or framing error (Check : RS485 bus need 120 Ohm terminating resistor or GND connection)
8: CRC error
9: Data lenght or content error
10: Modbus Exception (see Data)
11: LoRa downlink parameter error
12: LoRa uplink length error (Modbus data is truncated to maximal LoRa uplink length)

2.4 LoRa uplink payload structure on Port 100 (Setup Port)

Confirmed downlinks, with at least one payload byte, are answered with an uplink. The payload length is 5..21 bytes.

Byte No. [0...X]	Function/Meaning	Remarks
0	Comtac device type	Applications Type (0=LPN Modbus Standard, 1=LPN Modbus Easy)
1	Software version	Applications Main version
2		Applications Sub version
3	RSSI value	0..255 * -1 = RSSI [dB] (internal calculated with -139dB Offset)
4	SNR value	-128..+127 = +- Snr [dB] (signal-to-noise ratio)
5 + n		**Bits 0..3 REGn Error code (optional):** 0: No error 1: Not initialized 2: In use (busy) 3: No echo received -> Check RS485 Bus shorten 4: Function code not supported 5: No answer in the given time 6: Interframe timeout 7: UART parity or framing error (Check : RS485 bus need 120 Ohm terminating resistor or GND connection) 8: CRC error 9: Data length or content error 10: Modbus exception (Exception see Bits 4..7) **Bits 4..7 REGn Exception code (optional):** See http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

After the last configured register, no more data is sent, so the note (optional).

2.5 LoRa payload structure Port 101 LoRa configuration

2.5.1 Payload Downlink

In the case of a downlink, the first 3 bytes must be exactly right in order to trigger a configuration change:

Payload[0+1]: Module identifier 0x34 LSB und 0x05 MSB
Payload[2]: Applications type (1= LPN Modbus Easy)
Payload[3]: ConfigId:
0= All, following by data as in the uplink, whereby the shortened payload length also works
1=LazyDownlinkCnt (1 following data byte is expected)
2=ADR (1 following data byte is expected)
3=FrequencyPlan (1 following data byte is expected)
4=MinDR (1 following data byte is expected)
5=MaxDR (1 following data byte is expected)
6=DefDR (1 following data byte is expected)
7=Rx2DefDR (1 following data byte is expected)
8=RndTime (1 following data byte is expected)
9=SlotTime (2 following data bytes are expected, LSB first)
10=TimeSlotNr (2 following data bytes are expected, LSB first)
11=GrpDevAddr (2 following data bytes are expected, LSB first)
12=ConfirmedTx (1 following data byte is expected)
13=ConfirmedTries (1 following data byte is expected)
14=LivesignConfirmedTx (2 following data bytes are expected, LSB first)
15=RxConfirmTimeout (2 following data bytes are expected, LSB first)

16=DownlinkWatchdog (1 following data byte is expected)
17=ResetInterval (1 following data byte is expected)

Payload[4..] 1..23 data bytes with ConfigID = 0 otherwise 1 or 2 data bytes (LSB first)

If ConfigId> 0, further ConfigID + data blocks can follow.

The configuration is saved but only accepted by a reset -> use port 105.

2.5.2 Payload Uplink

Confirmed downlinks with at least one payload byte are answered with an uplink. The payload length is 27 bytes.

Payload[0]: Applications type (1=LPN Modbus Easy)
Payload[1]: Applications major version
Payload[2]: Applications minor version
Payload[3]: LoRa configuration changed by Port 101 downlink (0 oder 1)
Payload[4]: LazyDownlinkCnt (0 or 1)
Payload[5]: ADR_Off (0 or 1)
Payload[6]: FrequencyPlan (0:EU868_Default_3Ch 1:EU868_Semtech_8Ch 2:EU868_Standard_6Ch)
Payload[7]: MinDR (0..7)
Payload[8]: MaxDR (0..7)
Payload[9]: DefDR (0..7)
Payload[10]: Rx2DefDR (0..7)
Payload[11+12]: RndTime (0..9999s; LSB first)
Payload[13+14]: SlotTime (0..999ms; LSB first)
Payload[15+16]: TimeSlotNr (0..9999; LSB first)
Payload[17+18]: GrpDevAddr (0..9999; LSB first)
Payload[19]: ConfirmedTx (0 or 1)
Payload[20]: ConfirmedTries (1..8)
Payload[21+22]: LivesignConfirmedTx (0..9999; LSB first)
Payload[23+24]: RxConfirmTimeout (0..9999s; LSB first)
Payload[25]: DownlinkWatchdog (0..255h)
Payload[26]: ResetInterval (0..255h)

2.6 LoRa payload structure Port 105 Reset

2.6.1 Payload Downlink

Downlink with exactly 2 bytes, which must match in order to trigger a reset (necessary e.g. to take over port 101 configurations):

Payload[0]: 0x34
Payload[1]: 0x05

3 Configuration via USB interface

Insert the USB cable and open CFG.TXT, where all settings for LoRa and Modbus can be configured. **Configuration changes only take effect after a restart.**

3.1 LoRa configuration in CFG.TXT

```
LoRa (Vers. 0x43010200):
PrivateNetwork=0       // 1=Private 0=Public (overridden by DIP-SW-3 in REV00-HW)
LazyDownlinkCnt=0      // Downlink sequence counter is 0=checked 1=not checked (can be lower)

Activation:
OTA=0                  // 0=ABP used 1= OTA used (overridden by DIP-SW-2 in REV00-HW)
OTA(OverTheAir):
DevEUI=3734333665357D04
AppEUI=70B3D5FFFE297011
AppKey=2B8DEFCD2301674554761032DCFE98BA
ABP(ActivationByPersonalization):
FrequencyPlan=0 (0:EU868_Default_3Ch 1:EU868_Semtech_8Ch 2:EU868_Standard_6Ch)
DevAddr=0x00420136
NetwSesKey=1123456789ABCDEFFEDCBA9876543211
AppSesKey=EEDCBA98765432100123456789ABCDEE
Broadcast:
BC_Addr=0x00000000
BC_NetwSesKey=2223456789ABCDEEEEDCBA9876543222
BC_AppSesKey=DDDCBA98765432111123456789ABCDDD

Datarate (0.7;) DR_0... DR_7. SF12... FSK):
MinDR=0
MaxDR=7
DefDR=0 (Max.5 in OTA)// Depends also on FrequencyPlan
Rx2DefDR=3             // default receives data rate
ADR_Off=0              // ADR  (AdaptiveDataRate) is 0=on 1=off

Startup:               // Start-up behavior first sending and sendinterval restart (no restart when
                       // RndTime=1) in a time slot or random:
SlotTime=000 [100ms]   // for Var1 + 3 (min. 10 s at OTA; = 0-> OTA 10s ABP s = 2.3)
TimeSlotNr=0000        // Var1: (0 see Var2) 1.. 9999-> OTA: TimeSlotNr * 10s ABP: TimeSlotNr * 2.3s
RndTime=0001 [m]       // Var2: (0 see Var3) 1.. 9999-> randomize 10s... XXXX * 60s
GrpDevAddr=1024        // Var3: (0 see Var2 with 0060) 1..9999->TimeSlotNr=DevAddr/GrpDevAddr+1->Var1

Communication:
ConfirmedTx=0          // 0 = unconfirmed 1 = confirmed send (overwritten by DIP-SW-1 in REV00-HW)
ConfirmedTries=8 (1..8) // sets number of max. send retries in case of confirmedTX=1
LivesignConfirmedTx=0000 [m]  // Latest after this time+ConfirmedTxTimeout send confirmed uplink
ConfirmedTxTimeout=0000 [s]   // 0 = send immediate. x = no later than x seconds send
RxConfirmTimeout=0000 [s]     // 0 = confirm immediately. x = confirm after x seconds
DownlinkWatchdog=000 [h](0 not used) // 0..255 h after this time without Downlinks -> SW-Reset
ResetInterval=000 [h](0 not used)     // 0..255 h after first Uplink + this time -> SW-Reset
```

The startup parameters control the first transmission (uplink) and the following transmissions, which can be relevant for many Modbus bridges on same power supply in order to minimize simultaneous transmission:

- With the default settings (`RndTime` = 1), the first message (including joining with OTA) is triggered within one minute and the following messages in accordance with the `SendInterval`, whereby the following send intervals have the start time of the power on!
- If another startup setting is used, the `SendInterval` is restarted after first uplink. The start time of the further programs is tied to the first random transmission.
-

The first uplink can also be forced by buttons. Each further interval uplink varies randomly in the range of 0..2s. LivesignConfirmedTx ensures, at a defined interval, that the uplink is maintained by triggering a confirmed Tx. By means of ConfirmedTxTimeout, an application telegram can also be sent as confirmed if an application telegram is sent in this time window. The Acknowledgment can be terminated by means of the RxConfirmTimeout with the confirmed downlink, so an application response can also contain the acknowledgment during this time (the Ack is sent immediately at 0).

Transmission of data can be operated as confirmed (`ConfirmedTx=1)` or unconfirmed `(ConfirmedTx=0`) message(uplink). In case of confirmed uplink reception of a confirmation is expected. If non confirmation is received, transmission of the message is repeated until a confirmation is received or maximum number of transmissions is

reached. Maximum number of transmissions is specified by parameter "ConfirmedTries". If ConfirmedTries=1 the message is send only once, without any repetition, even in case of no confirmation received.
In order to increase the reception probability in case of a lack of confirmation, the data rate is reduced after every second transmission attempt.

Parameters "DownlinkWatchdog" and "ResetInterval" are intended for special applications only and should be kept 0 usually. "DownlinkWatchdog" >0 forces device reset after specified number of hours if there was no downlink message received during that time. "ResetInterval"> 0 forces device reset after spezified number hours since first uplink.

FrequencyPlan	*Channel*	*Frequency*	*Modulation / BW*	*Band*
EU868_Default_3Ch	0	868.100 MHz	MultiSF 125 kHz	1
	1	868.300 MHz	MultiSF 125 kHz	1
	2	868.500 MHz	MultiSF 125 kHz	1
Additionaly on EU868_Semtech_8Ch	3	867.100 MHz	MultiSF 125 kHz	0
	4	867.300 MHz	MultiSF 125 kHz	0
	5	867.500 MHz	MultiSF 125 kHz	0
	6	867.700 MHz	MultiSF 125 kHz	0
	7	867.900 MHz	MultiSF 125 kHz	0
	8 FSK	868.800 MHz	FSK 250 kHz, 50 kbps	2
	9 LoRa	868.300 MHz	SF7 250 kHz	1
Additionaly on EU868_Standard_6Ch	3	868.850 MHz	MultiSF 125 kHz	2
	4	869.050 MHz	MultiSF 125 kHz	2
	5	869.525 MHz	MultiSF 125 kHz	2
	8 FSK	868.300 MHz	FSK 250 kHz, 50 kbps	1
	9 LoRa	868.300 MHz	SF7 250 kHz	1
For all EU868 plans	RX2	869.525 MHz	SFx (see Rx2DefDR) 125 kHz	3

3.2 Modbus configuration in CFG.TXT

```
Modbus:
      Baudrate (00600..115200)
      |       Parity 0:None 1:Odd 2:Even 3:NoneExt(10Bits frame 8,1,NONE)
      |       | Mode 0:RTU 1:ASCII 2:RTU (2*Interdelay)
      |       | | Retries
      |       | | | Timeout [ms]
MBCfg=019200;0;0;1;500

Write FC=0x10 before each REG read:
WrRegAddr=00000 (0 not used)  // If> 0 execute for each REG read with respective DevAddr
WrRegData=00000               // Data (in BigEndian Format sent)
WrDelayToReadREG=5 [100ms]    // Waittime after a write before a read REG is submitted

SendInterval=0060 [m]         // Send-Intervall
MinSendOnChangeInterval=0010 [m] (0000 for none)
Register REG:
      DevAddr 0..255 0:not used
      |   Read-FC (1..4)
      |   | Addr
      |   | |     SendOnChange (send on interval and on 0:none 1:change)
REG00=011;4;00008;0
REG01=000;0;00000;0
REG02=000;0;00000;0
..
REG15=000;0;00000;0
```

A maximum of 16 registers can be configured, at least one device address (DevAddr) and a Modbus FunctionCode (FC) must be assigned.

Read-FC (Functioncode):
1: ReadCoils (16 coils)
2: ReadInputs (16 inputs)
3: ReadHoldingReg (16-bit register)
4: ReadInputReg (16-bit register)

Addr is the register address (register address = register number – 1).

Consecutive registers with same DevAddr, same FC and ascending Addr eg.:
REG03 = 011; 4; 00008;0
REG04 = 011; 4; 00009;0
REG05 = 011; 4; 00010;0
REG06 = 011; 4; 00011;0
are read out in a read command so that, for example, 8 bytes counter register can also be read correctly.

An uplink is generated each SendInterval or in case of SendOnChange additionally on change and expired MinSendOnChangeInterval. The SendInterval and MinSendOnChangeInterval are reloaded after each uplink.

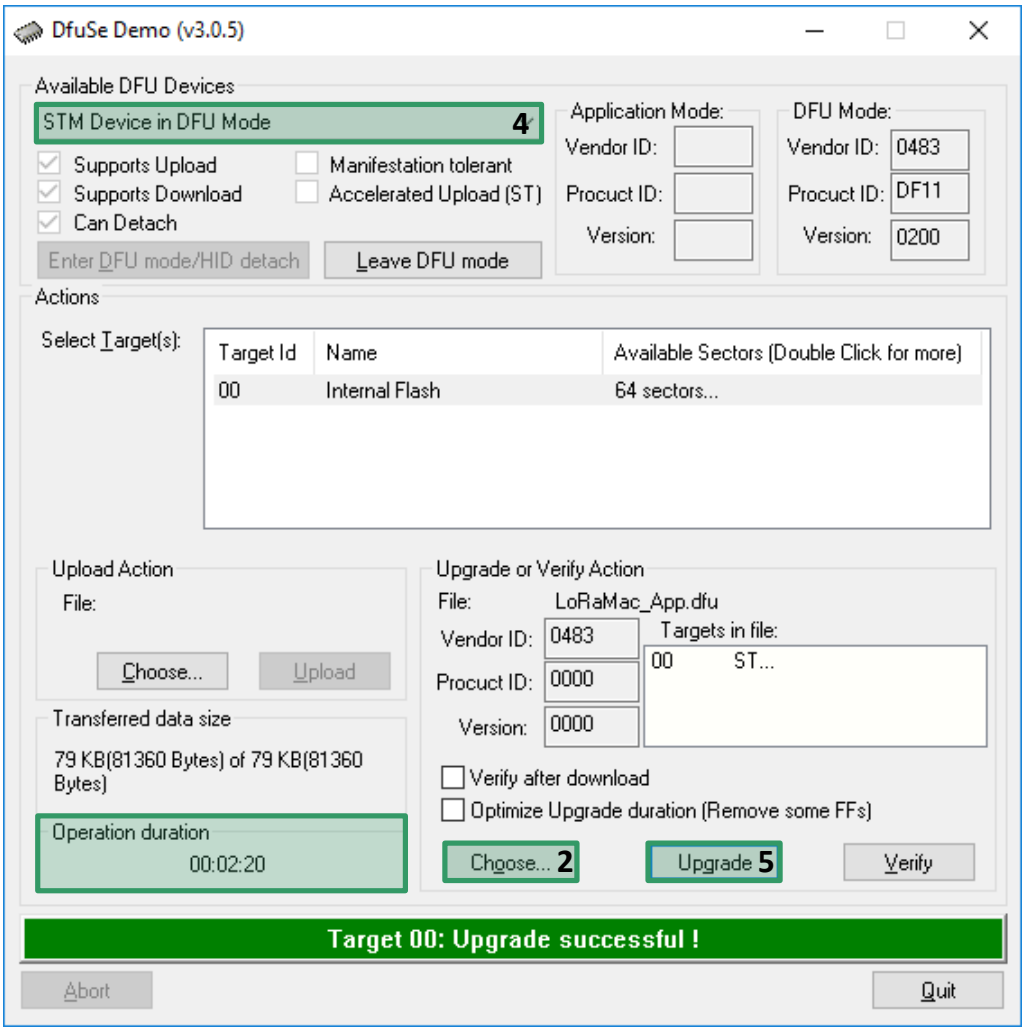
Register using SendOnChange will be read out continuously all other configured registers will be read out in the SendInterval. If SendOnChange register data changes and the MinSendOnChangeInterval is still going on, the register data can be overwritten several times, until the uplink happens.

When SendInterval is 0, all configured Modbus registers are read out continuously and the data can be requested by a confirmed Port 3 downlink.

4 Firmware update via USB bootloader (DFU Update)

Nodes which have a boot loader can be updated via USB-DFU.

1. Start up DFU Tool «DFuSe Demo»
   (Link → http://www.st.com/en/development-tools/stsw-stm32080.html).
2. Press "Choose..." button under **upgrade or verify action** (bottom right) to load the current DFU file.
3. Turn off device by removing supply and USB cable
4. Connect the USB micro plug to the PC using a cable, while holding down the “SEND” button.
5. Red Led should be flashing in half-sec-on-time and remaining LEDs should light -> Bootloader active.
6. The device is now in Bootloader mode (device appears under "Available DFU Devices").
7. Press «Upgrade» and ignore any messages. Updating takes about 2 minutes.
8. After the update, unplug the USB cable and restart the device.



IMPORTANT:
After installing the DFU Updater, note the DfuSe_en.CD00155676.pdf. On the first update, manually locate the driver path in "C: \ Program Files (x86) \ STMicroelectronics \ Software \ DfuSe v3.0.5 \ Bin \ Driver \".