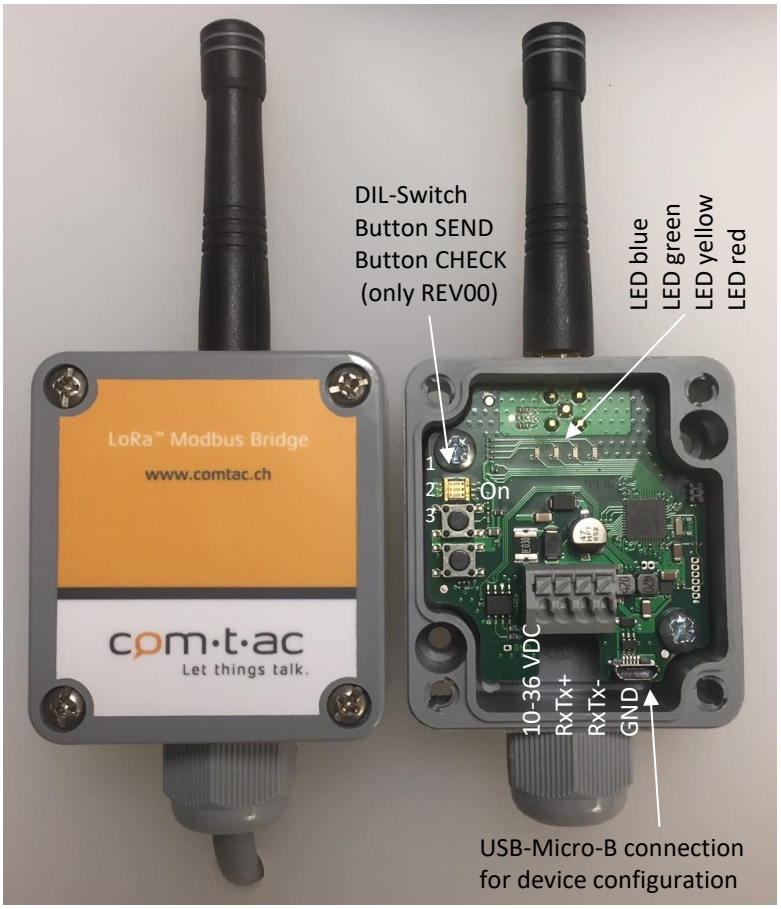
comtac AG
Allenwindenstrasse 1
CH-8247 Flurlingen
info@comtac.ch
www.comtac.ch
T +41 52 647 30 30

LPN Modbus Bridge / Specification V0.26-1

Comtac AG
CH-8247 Flurlingen



History

Date	Description	Issue of document	Firmware version
2016-12-14-Kd	REV00 V0.00 First Release		
2016-12-15-Kd	REV00 V0.01: - Corr. Description toDPStdMbFormat, corrected Lenghtbyte + 0x80 SW V0.01 - Port uplink extended with RSSI and Snr		
2017-01-04-Kd	REV00 V0.02: New “Write FC 0x10” before Read DP + new Mode 3 configurations		
2017-01-19-Kd	REV00 V0.03: Changed response timeout `Timeout [ms] (max. 635)` to max. 999ms		
2017-01-31-Kd	REV00 V0.04: Extended DIP-SW3 with Public/Private networking		
2017-03-14-Kd	REV00 V0.09: Added a separate format for value list in extended DP Mode uplink		
2017-03-31-Kd	REV00 V0.11: New Port 100 application type query		
2017-05-10-Kd	REV00 V0.12: Implemented USB-CDC function, added Public/Private choice per CFG.TXT		
2018-01-31-Kd	REVxx V0.16: New HW Rev01 (without DIL-Switch, with 16MHz Quartz)		
2018-04-11-Kd	REVxx V0.17: New FrequencyPlan (corr. ConfirmTx in Firmware for REV01)		
2019-01-10-Kd	REVxx V0.17: New Sendmethod Eventtrigger		
2019-11-29-Kd	REVxx V0.22: Baudrate limits 600..115200 (old Version could got Resets with > 57600Baud)		
2020-03-25-Kd	REVxx V0.23: Transceiver Uplink Payload -> use SlaveAddress for DevAddr (was fix 0)		
2020-04-29-Kd	REVxx V0.24: FC1 + FC2 were reversed		
2020-10-26-Kd	REVxx V00.25: after startup time measure interval will be newly restarted (RndTime is per Default 1, when needed set **RndTime to 1 after SW update)** and new Config. DownlinkWatchdog + ResetInterval		
2020-11-25-Kd	REVxx V00.26: New Ports 101 LoRa-Config., Port 105 Reset and new config.param. ConfirmedTries, in the case of confirmed uplinks, the payload is shortened depending on the ADR and MIN_DR, so that the last repetition would also work with reduced DR. **RndTime is automatically changed to 1 after the update, if old App. Version <0.26 and RndTime was 10.** **With RndTime 1, no DP interval restart after the first uplink will be done.** In the case of DPExt = 0 and DPStdMbFormat = 0, the payload size is used 3 bytes higher if necessary, old		V0.26

	versions have max. Payload size always calculated based on the active DPStdMb format.		
2021-03-22_Su	Description of parameter `ConfirmedTx included`	E1332-LoRa_Modbus_Bridge_EN_V0.26-1.docx	V0.26

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 V1.0.1 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.

1.3 Function of the 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 data points (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 Configuration via USB interface

Insert the USB cable and open CFG.TXT, where all settings for LoRa and Modbus can be configured (not working in USB-CDC mode). **Configuration changes only take effect after a restart.**

2.1.1 LoRa configuration in CFG.TXT

```
LoRa (Vers.0x43010200) :
PrivateNetwork=0       // 0= Public   1= Private (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     // 0 for not used
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 receive data rate
ADR_Off=0              // ADR  (AdaptiveDataRate) is 0=on 1=off

Startup:               // Start-up behavior first sending and DP-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*10 s ABP: TimeSlotNr*2.3 s
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 uplink (overwritten by DIP-SW-1 in REV00)
ConfirmedTries=8 (1..8)        // Every 2nd attempt reduces the DR to MinDR (payload reduction)
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=0000 [m](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, with RndTime = 1, the first message (including joining with OTA) with read data points is triggered within one minute and the following messages in accordance with DP send interval (DP for data points), whereby the following send intervals have the start time of the power on!
- If another startup setting is used, the last DP values are sent in the first transmission and the DP measurement and transmission intervals are restarted, which triggers a new transmission with the current DP values. Thus, the start time of the further transmission 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.

Using the `DownlinkWatchdog`, a software restart can be triggered after the first uplink (except JoiningRequest) in the event of downlink failures (timeout) and thus reinitialized to the default Rx2 setting. With `ResetInterval`, a software reset is always triggered after first uplinks plus the configured hours. After a software reset, the first uplink is triggered in 10..20s (no measurement interval shift is inserted).

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

2.2 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
      |       | | Retries
      |       | | | Timeout [ms]
MBCfg=019200;0;0;1;500

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

SendInterval=0001 [m]         // Can be used as a specific transmission interval on the DPs
DPStdMbFormat=0               // 1=DevAddr+FC+Length are sent in the DP command block uplink
Datapoints DP:
      DevAddr 0..255 0:not used
      |   Interval 0:5m 1:15m 2:30m 3:1h 4:4h 5:12h 6:1D 7=SendInterval
      |   | Read-FC (1..4)
      |   | | Cnt (max 32 for Reg.)
      |   | | |  Addr
DP_00=011;7;4;02;00008
DP_01=000;0;0;01;00000
DP_02=000;0;0;01;00000
..
DP_15=000;0;0;01;00000
```

A maximum of 16 data points can be configured, for that at least one device address (`DevAddr`), a Modbus FunctionCode (`Read-FC`) and a number of registers (`Cnt`) must be assigned.

Read-FC :
1: ReadCoils (Cnt=number ofc)
2: ReadInputs (Cnt=number of inputs)
3: ReadHoldingReg (Cnt=number of 16Bit Register)
4: ReadInputReg (Cnt=number of 16Bit Register)

The interval (`Interval`) forms the measurement and transmission interval for each data point. For a freely configurable interval, the general transmission interval (`SendInterval`) can also be used. The register address (`Addr`) is the final used address in the Modbus telegrams (register address = register number – 1).

2.2.1 Extended Modbus configuration in CFG.TXT

```
DPExt=0          // Extended data point format
Extended DP (used when DPExt is 1):
      Measureinterval 0:Interval 1:loop 2:1m 3:5m 4:15m 5:30m 6:1h 7:4h
      | Sendmethode 0:Value 1:Mean 2:Min. 3:Max. 4:Trigger 5:Triggerevent
      | | Datatype 0:INT16 1:UINT16 2:INT32 3:UINT32 4:IEEE754_SP
      | | | Byteorder 0:ABCD 1:CDAB 2:DCBA 3:BADC
      | | | | Triggerinterval (see Interval)
      | | | | | Trigger 0:Lower 1:Upper 2:Both
      | | | | | | Lower level (see Datatype dec 0 or hex 0x or float 0.0)
      | | | | | | |             Upper level
DPe00=0;0;0;0;0;0;            0;            0
DPe01=0;0;0;0;0;0;            0;            0
DPe02=0;0;0;0;0;0;            0;            0
..
DPe15=0;0;0;0;0;0;            0;            0
```

The 16 standard data points can be provided with an extended task. The task (`Sendmethode`) can be a value list, a value averaging, a peak value function (Min-Max) or a threshold value recognition due to the faster measurement interval (`Measureinterval`). For this, the data type (`Datatype + Byteorder`) must be defined.
For threshold detection (`Trigger + Triggerevent`), the threshold values (`Lower+Upper level`) must also be defined and if these values are exceeded, the trigger interval (`Triggerinterval`) is used adaptively instead of the standard interval (`Interval`) as the send interval. With `Trigger`, the time interval that complies with the respective intervals is adhered to, and at `Triggerevent` after threshold exceeded and expiration of the `Triggerinterval` (since last exceeded) immediately an uplink is issued. With the threshold value recognition the mean value for `Trigger` or the last values for `Triggerevent` is sent with continuous measuring (loop), otherwise a value list of the last measurements is sent.
A maximum of 24 measurements per DP are recorded and used for the evaluation (mean value, min. and max.).

The `Measureinterval` specifies the Modbus register query interval, and the query data is converted to a query reading using the `Datatype` and `Byteorder`. With `Sendmethode = Trigger + Triggerevent`, the interrogation readings are then immediately checked for `Lower`- and `Upper level` and, if exceeded, the adaptive Trigger Interval is activated. The interrogation reading is then stored in a maximum of 24 measured values of a measured value buffer, the oldest measured value being overwritten when the buffer is full. When the `Sendinterval` (or adaptively the `Triggerintervall`) is reached, a value list, the mean, minimum or maximum value is taken from the measured value buffer depending on the send method.

A value list is only sent with `Sendmethode = Value` and with `= Trigger + Triggerevent` with `Measureinterval <> loop`.

If the extended Modbus variant is activated by means of `DPExt=1`, the LoRa server should provide the devices with the time of day via Port 1 Broadcast (Multicast) every minute, since this is used in the data uplink. After switching on or resetting, the internal time is set to midnight 1.1.2016.

3 LoRa up- and downlinks

Commands from the server to the node (LPN Modbus 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.

All data is sent in Little Endian format (LSB first) (DCBA), except direct Modbus register values.

If not all data points fit into an uplink, the other DPs are sent after TimeOnAir timeout. If the uplinks no longer follow on with the data, especially at data rates of SF12-SF10, the higher DPs (15 and down) are no longer transmitted.

3.1 LoRa payload structure Port 1 Time + application version

Time is used with extended data point format as the elapsed time since midnight. The date is not used.

3.1.1 Downlink set time

Payload[0..3] : Unixtime (LSB first)

3.1.2 Uplink read time, application version and RSSI+Snr

Confirmed Downlinks, mit mindestens einem Payload Byte, werden mit einem Uplink beantwortet.

Payload[0..3] : Unixtime (LSB first)
Payload[4] : 0=Unixtime was set during last day 1=Unixtime not set during last day
Payload[5] : Application major version
Payload[6] : Application minor version
Payload[7] : 0..255 * -1 = RSSI [dB] calculated with a -139dB offset
Payload[8] : -128..+127 = +- Snr [dB] RSSI [dB] calculated with a -139dB offset

3.2 LoRa payload structure Port 2 Modbus

Command 1			(Command 2)			(Command n)		
Startbyte	(Lenghtbyte)	(Data)	Startbyte	(Lenghtbyte)	(Data)	Startbyte	(Lenghtbyte)	(Data)

The payload is divided into command blocks, each command must contain at least the start byte.

The start byte in turn contains a command identifier (mode), which describes the further evaluation.
Startbyte Bit 6+7: Mode (0..3)
0: DP-Data (Data points data)
1: DP-Configuration (Configure data points over LoRa)
2: Transceiver (LoRa-Modbus bridge) (**must be used as a single or last command**)
3: Configuration (e.g. configure RS485 parameters)

Optionally, depending on the mode, a length byte (number of the following data bytes) and the data can follow. The start byte itself may contain further information.

For the uplinks, only mode 0 (DP data) can contain several command blocks.

Command block byte = KB [x] corresponds to the index in the block.

3.2.1 Mode 0 DP-Data downlink (without lenght+data)

Startbyte KB[0]:
Bit 6+7: Mode (0..3) = 0
Bit 2..5: DPNo (0..15) Data point number
Bit 0+1: not used

The server can request the re-sending of the desired data point by means of this command.

3.2.2 Mode 0 DP-Data uplink DPExt=0

Startbyte KB[0]:
Bit 6+7: Mode (0..3) = 0
Bit 2..5: DPNo (0..15) Data point number
Bit 1: Errorstatus flag
Bit 0: 0 (not extended format)
Lenghtbyte KB[1] Bit0-6: Number of following data bytes, bit 7 -> active if DPStdMbFormat = 0
Data KB[2 .. (2+ KB[1])]:
If the errorstatus flag is set with length=2-> **Modbus-Error command block**:

Startbyte=b00XX'XXI0	Lenght=2	DevAddr	Errorcode

Errorcodes:
1: Not initialized
2: In use
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

If the errorstatus flag is set with length=3 -> **Modbus-Exception command block**:

Startbyte=b00XX'XXI0	Lenght=3	DevAddr	0x80 + FC Functioncode	Exceptioncode

Exceptioncode see http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

Errorstatus flag = 0 is followed by valid DP values -> **Modbus DP command block with DPStdMbFormat=0**:

Startbyte=b00XX'XX00	**0x80 +** Lenght=n	n Databytes

Attention: For the length, the highest bit (bit 7) must be masked out.

Errorstatus flag = 0 is followed by valid DP values -> **Modbus DP command block with DPStdMbFormat=1**:

Startbyte=b00XX'XX00	Lenght=3+n	DevAddr	FC Functioncode	Length n Databytes	n Databytes

Databytes correspond to the answer see http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

3.2.3 Mode 0 DP-Data uplink DPExt=1 (extended format)

In the extended format a timestamp is included, this timestamp (2 bytes LSB first) contains the past seconds (4s interval) since midnight. The time stamp is supplied per telegram at the first occurrence or change, otherwise a previously supplied flag, the ST bit (SameTime bit), is set to 1 to indicate the absence of the following 2 bytes for the time stamp. **The uppermost bit (Bit15) in the timestamp indicates that the time has not been set by Port 1 command for at least 1 day.**

Startbyte KB[0]:
Bit 6+7: Mode (0..3) = 0
Bit 2..5: DPNo (0..15) Data point number
Bit 1: Errorstatus flag
Bit 0: 1 (Extended data format)
Lenghtbyte KB[1]: Number of the following data bytes
Data KB[2 .. (2+ KB[1])]:
If the errorstatus flag is set -> **Modbus-Error command block**:

Startbyte=b00XX'XX**II**	Lenght=2+(2)	Errorstatus+ST	(Timestamp)	Error- or Exceptioncode

Errorstatus Bit 1-7: 0=ok 1=Parameter error 2=Modbus Errorcode follows 3=Modbus Exceptioncode follows
ST-Bit 0: 0=Timestamp follows 1=like last timestamp (Timestamp is omitted)
(Timestamp): 2Byte timestamp
Errocodes or Exceptioncode

Errorstatus flag = 0 is followed by valid DP values -> **Modbus DP-Extended command block**:

Command header		DP (with same timestamp)			(Further DPs)
Startbyte=b00XX'XX**0**I	Lenght=x **(< 0x80)**	numVal+valSize +ST	(timestamp)	DP-Data	…

DP (with same timestamp):
numVal Bit 4..7: 1..15=Ext.DP number of data with (valSize + 1) bytes data size 0=Standard DP
valSize Bit 1..3: 0..7 + 1= Data point size in bytes (with standard DP always 1)
DP-Data: With standard DP the data follows and with ext. DPs numVal DPs with the same time stamp follows.

Errorstatus flag = 0 is followed by valid DP values -> **Modbus DP-Extended command block value list**:

Command header		DP (from timestamp with the same measure interval)				(Further DPs)
Startbyte=b00XX'XX**0**I	**0x80** + Lenght=x	numVal+valSize +ST	(timestamp)	Measure interval	DP-Data	…

DP (Attention: For the length, the highest bit (bit 7) must be masked out):
numVal Bit 4..7: 1..15=Ext.DP number of data with (valSize + 1) bytes data size 0=Standard DP
valSize Bit 1..3: 0..7 + 1= Data point size in bytes (with standard DP always 1)
Measure interval: 0..255 [Minutes] 0= Loop
DP-Data: Following numVal ext. DPs (n=1..15 DP-Zeit[s] = (timestamp & 0x7fff)*4+(n-1)*60*Measureinterval).

3.2.4 Mode 1 DP-Configuration downlink

Startbyte KB[0]:
Bit 6+7: Mode (0..3) = 1
Bit 2..5: DPNo (0..15) Data point number
Bit 1: Reset-DP 0:Set-DP 1:Reset-DP (no further command bytes), sets device address to 0
Bit 0: not used
Lenghtbyte KB [1]: Number of the following data bytes (only with DP-Configuration Set-DP or when reading = 0).
Data KB[2 .. (2+ KB[1])]: (only with DP-Configuration Set-DP)

Startbyte=b01XX'XX0Y	Lenght>=0	(configuration data)	(ext. config.)	(ext. config. threshold data)

For confirmed downlink and length 0 read configuration, see Mode 1 DP-Configuration uplink
Standard DP configuration data Blockbyte[2..7]:
KB[2] Methode Bit5..7: 0=Value(list) 1=Mean 2=Min. 3=Max. value 4=Trigger
Interval Bit0..4: 0:5m 1:15m 2:30m 3:1h 4:4h 5:12h 6:1D 7= SendInterval (see config.)
KB[3] Functionscode FC-Read: 1:ReadCoils 2:ReadInputs **3:**ReadHoldingReg. 4: ReadInputsReg.
KB[4] Number of bits (at FC=1 or 2) or number of 16-Bit Register (FC=3 or 4)
KB[5+6] Data address (LSB first)
KB[7] Device address 1..254
Optional ext. DP configuration data (if method> 0 or measure interval for value list required):
KB[8] Measure interval Bit5..7: 0:Interval (Bit3-5) 1:loop 2:1m 3:5m 4:15m 5:30m 6:1h 7:4h
Trigger Bit3+4: 0=Lower 1=Upper 2=Lower+Upper
Trigger send interv. Bit0..3: 0:5m 1:15m 2:30m 3:1h 4:4h 5:12h 6:1D 7=SendInterval (see config.)
KB[9] Data type Bit4..7: 0:INT16 1:UINT16 2:INT32 3:UINT32 4:IEEE754_SP

Byte order Bit0..3: 0:ABCD 1:CDAB 2:DCBA 3:BADC (ABCD BigEndian) (DCBA LittleEndian)
Optional Threshold Data (required for trigger method) (format like byte order):
for data type <2 (16Bit):
KB[10+11] Lower Level
KB[11+12]: Upper Level
for data type >=2 (32Bit):
KB[10..13]: Lower Level
KB[14..17]: Upper Level

3.2.5 Mode 1 DP-Configuration uplink

At confirmed downlink at least the mode + DPNr will return on port 2. If the data length is not correct, a 0x41 is sent back to signal a parameter error.
If the uplink was made with the length 0, the configuration data are read out. **Only one (the last) configuration can be sent in an uplink telegram**. A device address with the value 0 means that the DP is not used.

Startbyte [0]:
Bit 6+7: Mode (0..3) = 1
Bit 2..5: DPNo (if no parameter error)
Bit 0+1: Errorstatus 0=ok 1= parameter error
Data [1.. Payloadend]: (optional data structure see Mode 1 DP-Configuration downlink)

3.2.6 Mode 2 transceiver downlink (last command block)

Startbyte KB[0]:
Bit 6+7: Mode (0..3) = 2
Bit 0..5: not used
Data KB[1..Payloadend]:

Startbyte=b10XX'XXXX	DevAddr	FC Functioncode	Modbus data

Modbus data acc. http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

3.2.7 Mode 2 transceiver uplink

If downlink confirmed is sent, an uplink response follows.
Startbyte [0]:
Bit 6+7: Mode (0..3) = 2
Bit 2..5: not used
Bit 0+1: Errorstatus 0=ok 1= parameter error 2=Modbus Errorcode follows 3=Modbus Exceptioncode follows
Data [1..Payloadend]:
If Errorstatus = 1 -> no data.
If Errorstatus = 2 -> 1 byte of data with the Errocode.
If Errorstatus = 3 -> 1 byte of data with the Exceptioncode.
If Errorstatus = 0 :

Startbyte=b10XX'XX00	DevAddr	FC Functioncode	Modbus data

Modbus data acc. http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

3.2.8 Mode 3 configuration downlink

Startbyte KB[0]:
Bit 6+7: Mode (0..3) = 3
Bit 2..5: ConfigCmd 0=RS485Bus 1=SendInterval 2=DPExt 3=Write FC 0x10
Bit 0+1: not used
Length KB [1]: Number of the following data bytes. **At 0 and confirmed downlink read out data with uplink.**
Data KB[2 .. (2+ KB[1])]: see below

3.2.8.1 RS485-Bus configuration

Startbyte=b1100'00xx	Lenght=0 or >=9	(Configuration data)

Configuration data:
KB[2] Bit 1+2: Parity 0:None 1:Odd 2:Even 3:NoneExt(10Bits RTU frame)
Bit 0: Mode 0:RTU 1:ASCII
KB[3..6]: Baudrate (LSB first)
KB[7]: Retries (max. 9)
KB[8+9]: Max. Response timeout [ms] (LSB first)

3.2.8.2 SendInterval (general send interval)

Startbyte=b1100'01xx	Länge=0 or >=2	(Configuration data)

Configuration data:
KB[2+3] SendInterval [m] (0..9999) (LSB first)

3.2.8.3 DPExt (use extended data point format)

Startbyte=b1100'10xx	Lenght=0 or >=1	(Configuration data)

Configuration data:
KB[2] DPExt 0 or 1

3.2.8.4 Write FC 0x10 (Write before each Read DP)

Startbyte=b1100'11xx	Lenght =0 or >=2	(Configuration data)

Configuration data:
KB[2+3] WrRegAddr (0 for not used)
KB[4+5] WrRegData (Bytes are swept again when sending -> BigEndian)
KB[6] WrDelayToReadDP (0..9 [100ms])

3.2.9 Mode 3 configuration uplink

A confirmed downlink, will be answered by an uplink.

Startbyte [0]:
Bit 6+7: Mode (0..3) = 3
Bit 2..5: ConfigCmd 0=RS485Bus 1=SendInterval 2=DPExt und 3=Write FC 0x10
Bit 0+1: Errorstatus 0=ok 1= parameter error
Data [1..Payloadend]: see at the downlinks.

3.3 LoRa payload structure Port 100 applikations type, version, RSSI+Snr and Errorcodes

3.3.1 Uplink

Confirmed downlinks, with minimum one payload byte, will be answered by an uplink.
The payload length is 5..21 bytes.
Payload[0]: Applications type (**0=Standard**; 1= Easy)
Payload[1]: Applications major version
Payload[2]: Applications minor version
Payload[3]: 0..255 * -1 = RSSI [dB] (calculated with -139dB offset)
Payload[4]: -128..+127 = +- Snr [dB] RSSI [dB] (calculated with -139dB offset)
Payload[5+n]: Bits 0..3 DPn Errorcode (optional): see Errocodes
Bits 4..7 DPn Exceptioncode (optional): see Exceptioncode

After the last configured register, codes will no longer be sent, hence the note (optional).

3.4 LoRa payload structure Port 101 LoRa configuration

3.4.1 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 (0=Standard)
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.

3.4.2 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 (0=Standard)
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)

3.5 LoRa payload structure Port 105 Reset

3.5.1 Payload Downlink

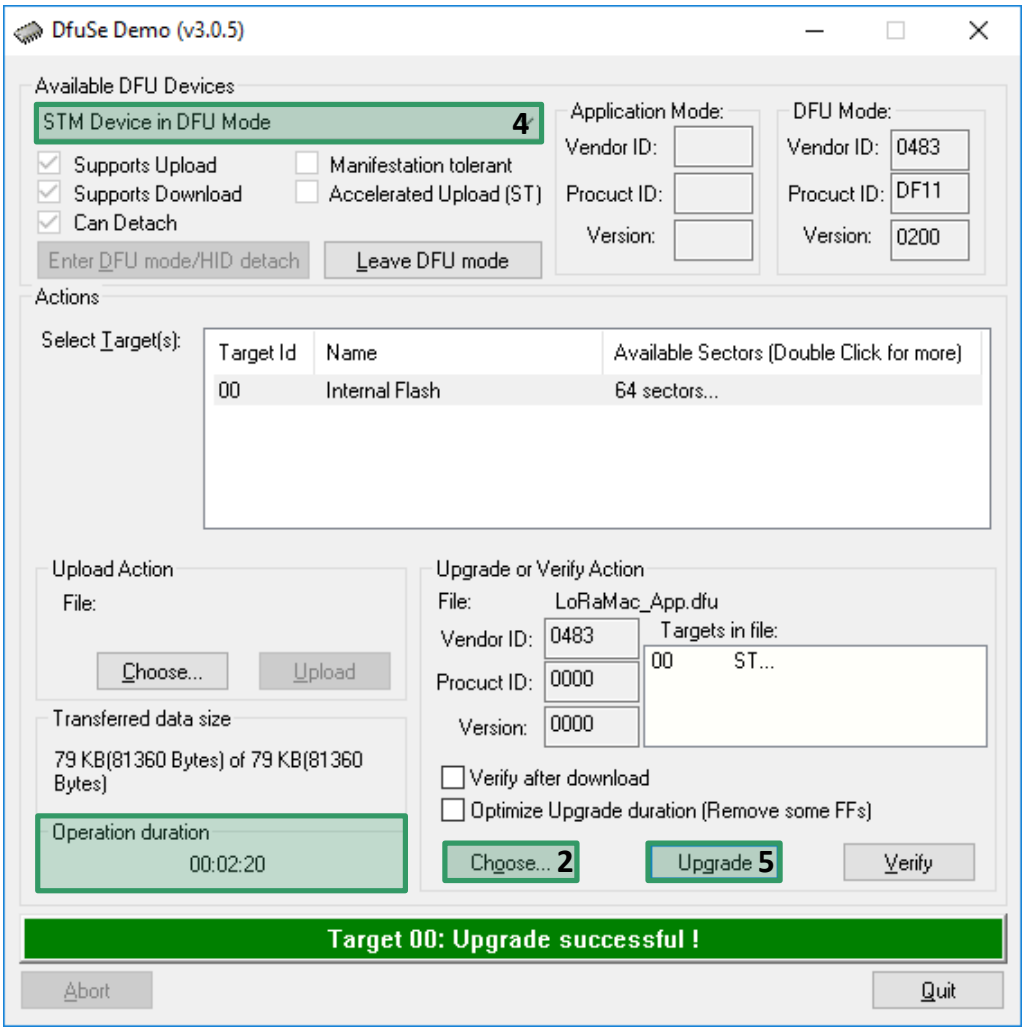
Downlink with exactly 2 bytes, which must match in order to trigger a reset (necessary e.g. to take over port 101 configurations):

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

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 \".