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

Cluey AM LR

Operating manual

2 Content

1 Document History

Version	Date	Notes/Changes
2.00	24.03.2023	1st released engl. version 2.0
		Error corrections

2 Contacts

2.1 Manufacturer

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

2.2 Distribution

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

Your sales and contact partner:

2.3 Technical support

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

3 General notes

This documentation is an integral part of the product and should be kept throughout its service lifetime and passed on to subsequent users of the product.

This documentation has been prepared with great care. Nevertheless, errors cannot be completely avoided.

We are grateful for any information about errors or suggestions. →support@comtac.ch

3.1 Validity

This documentation applies to the following product and product accessories:

Designation	Version
Cluey AM	Hardware version: V02
Firmware version	3.02
Payload version	3
Payload decoder version	00.07 (Cluey_KM_AM_TM_Decoder_V00.07.js)

3.2 Copyright

3.3 Symbols

In this manual, the following symbols are used to indicate information relating to the proper and safe use of the device.

 WARNING	Failure to comply with the instruction may result in death or serious injury.
 ATTENTION	Failure to comply with the instruction may result in injury or property damage.
 NOTE	Failure to follow the instructions may result in damage to the device.
 TIP	This symbol indicates instructions for optimum operation, optimal settings, easier handling and avoidance of errors.
 REFERENCE	This symbol indicates references to further or supplementary information.

4 Intended use

This product is intended for use in automation systems, switchgear, electrical installations or as a stand-alone device for the acquisition of digital and analogue signals, for the output of digital signals coupled to higher-level systems via radio link. It is intended in particular for use in medium-voltage transformer stations and similar installations.

Installation, commissioning and maintenance may only be carried out by appropriately trained and qualified staff and after reading and understanding these operating instructions.

The product may only be used at operating locations for which the degree of protection specified in the technical data is sufficient and the specified operating environmental conditions are guaranteed. Use in potentially explosive atmospheres is not permitted. The implementation of safety and protection monitoring functions (such as emergency OFF) is not an intended use of the product.
Especially in the case of radio technologies, the legal guidelines vary from country to country.
The product may only be operated in regions, with settings and with antennas for which compliance with the guidelines is ensured.

The operator is responsible for compliance with the guidelines.

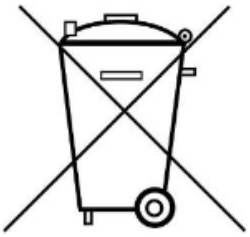
5 Safety instructions

The following instructions must be observed:

	When opening the housing, make sure that no liquid/moisture (rain) or dirt gets into the housing. Do not touch any electronic parts.
	Only use suitable accessories. Only components that ensure SELV/PELV isolation are allowed to be connected to the terminals.
	Never operate the device without the antenna connected and only with a suitable antenna to avoid damage to the device and to ensure compliance with the relevant guidelines.
	Lightning protection and earthing measures may have to be taken for the antenna to prevent dangerous overvoltages and leakage currents that can damage the device itself and connected devices.
	Use only propanol to clean the housing, contacts and circuit board. For cleaning, switch off external supply voltage and remove internal battery. Do not use contact spray!
	Replace defective or damaged devices.
	Avoid electrostatic discharge! Electronic components are used in the device which can be damaged or destroyed by electrostatic discharge if touched. Pay attention to safety measures against electrostatic discharge when connecting, opening the device and especially when replacing the battery.
	Avoid reverse polarity and excessive voltages. Before commissioning, check the correct wiring of the connections and correct positioning of the plugs. Do not connect live wires to the antenna connection.
	Battery replacement: Only use batteries of the same type as the battery supplied (see also technical data). Make sure that the polarity is correct.
	The device contains primary batteries containing lithium. They must not be (re-)charged. Lithium-containing (Li-) batteries are safe if handled properly. If used and stored improperly, lithium-containing batteries can cause fires. Do not use defective, damaged, deformed or inflated batteries. Batteries (even damaged ones) do not belong in household waste. Dispose of used batteries appropriately!
	Use only suitable accessories. Prefer those supplied and recommended by us. Other accessories may impair appliance safety and proper functioning.

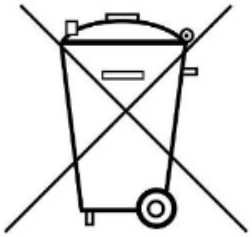
6 Disposal

6.1 Device disposal



Devices with electrical components must not be disposed together with household waste. They must be collected separately with electrical and electronic waste in accordance with local regulations and the laws currently in force.

6.2 Battery disposal



In connection with the sale of batteries or the delivery of devices containing batteries, we are obliged to inform you of the following:

Batteries must not be disposed with household waste. You can return used batteries to a municipal collection point or to your local retailer. As a distributor/manufacturer of appliances containing batteries, we are also obliged to take back used batteries, although our obligation to take back used batteries is limited to those that we carry or have carried in our range as new batteries. You can therefore return used batteries of the afore mentioned type either to your sales partner or to the manufacturer with sufficient postage.

Please note the above instructions.

7 Abbreviations

DC	Direct current
SMA	Sub-miniature A, high-frequency plug/socket
$LiMnO^2$	Lithium Manganese Oxygen, Battery Type
USB	Universal Serial Bus, serial interface
EMC	Electromagnetic Compatibility
RED	Radio Equipment Directive
RoHS	Restriction of (the use of certain) Hazardous Substances in electrical and electronic equipment
UTC	Universal Time Code, time format
GPS	Global Positioning System
ADR	Adaptive Data Rate, LoRaWAN® network function comply with LoRaWAN® specification
OTAA	Over The Air Activation, registration procedure for LoRaWAN® terminals on the LoRaWAN® network
ABP	Activation By Personalization, registration procedure for LoRaWAN® nodes
EUI	Extended Unique Identifier, worldwide unique identifier
DevEUI	Device Extended Unique Identifier, unique identifier for LoRaWAN® terminals
MSB	Most Significant Byte
LSB	Least Significant Byte
MSB	Most Significant Byte e.g. Bit15 to Bit07 of a 16Bit word
LSB	Least Significant Byte e.g. Bit07 to Bit00 of a 16Bit word
MSW	Most Significant Word e.g. Bit32 to Bit16 of a 32Bit Word
LSW	Least Significant Word e.g. Bit15 to Bit00 of a 16Bit word

8 Technical specification

8.1 Technical data

Power supply	Values	Note
Power supply ext.	5 ... 32VDC	
Power consumption ext.	0.3W	Typical
Inrush current	up to 2A	
Internal battery	3V Lithium primary battery, C-cell Type: CR26500 3.0V	Operating time depending on configuration
Signal connections 1...8		
Configurable functions of the signal connections	Digital input (low power)	IO 1-4 IN 5-8
	High level digital input	IO 1-4 IN 5-8
	Analogue input 0...10V	IO 1-4 IN 5-8
	Analogue input 0...20mA	IO 1-4 IN 5-8
	Digital output	IO 1-4
Dielectric strength	>= 0V; <=32VDC	
Digital input (low power)		
Switching threshold	Low: < 1.5V; high: open input or voltage > 2.5V	Input in this configuration is designed for contacts switching to GND
Input current	-1mA, pulsed, per input	
High Level Digital Input		
Switching thresholds	Low: < 5V; High: > 8.5V (high level)	Input is provided for external switching voltage
Input current	<10uA	
Counter size	24 bit	
Max. pulse counting frequency	Battery operation: 2 Hz Ext. voltage supply: 10Hz	Depending on the configuration
Overflow pulse counter	16'777'216	Overflow is visible in the status bit of the input
Operating time counter, resolution	1 s ,1min,1 h	
Overflow operating time counter	every 0.5 years Operating time	Overflow is recognisable in the status bit of the counter value of the respective input
Analogue input 0...10V		
Measuring range	0...12.5 V	Scaling: 0...10V corresponds to 0...10000
Input resistance		
Overflow / sensor error detection	>10.5V	If this value is exceeded, the "overflow" bit is set in the status of the measured value
Invalid- / open-circuit detection	< 1.8V	If this value is not reached, the "invalid" bit is set in the status of the measured value.
Resolution	12 Bit	

Measurement error	<0.5% of the measuring range	
Analogue input 0...20mA		
Measuring range	0...25mA	Scaling: 0...20mA corresponds to 0...10000
Input burden	200 Ω	
Overflow / sensor error detection	>10.5V	If this value is exceeded, the "overflow" bit is set in the status of the measured value
Invalid- / open-circuit detection	< 1.8V	If this value is not reached, the "invalid" bit is set in the status of the measured value.
Resolution	12 Bit	
Measuring error	<0.5% of the measuring range	
Outputs 1...4		
Output voltage (high)	For battery operation: 15V with external supply voltage: equal to supply voltage with USB supply: 15V	Output is active high switching
Output current (high)	USB/Ext. DC: 50mA per output, all outputs together max. 200mA Battery: 50mA per output, all outputs together max. 50mA max. 50mA	
Display & controls		
Button	Keystroke < 5s: Send telegram Keystroke > 5s: (re-)join Keystroke > 10s: Device reboot	See description key
LED	"USB", green: - ext. USB "DC IN", green: - ext. supply voltage "LED A", orange: Error status (e.g. configuration) "LED B", orange: - Joining/joined /transmitting	LEDs are only permanently activated with external power supply
Connections		
Antenna	50 Ohm, SMA	Int. antenna included
Inputs & ext. supply	Pluggable, 0.14 - 0.5 mm^2	
Configuration	Micro USB	
Radio interface		
Technology	LoRaWAN®	
MAC version	1.0.3	
Operating mode	Class C - with external supply Class A with battery operation	
Application procedure	OTAA APB	
Frequency band	EU868, Rev A	
Max. transmitting power	+14dbm	

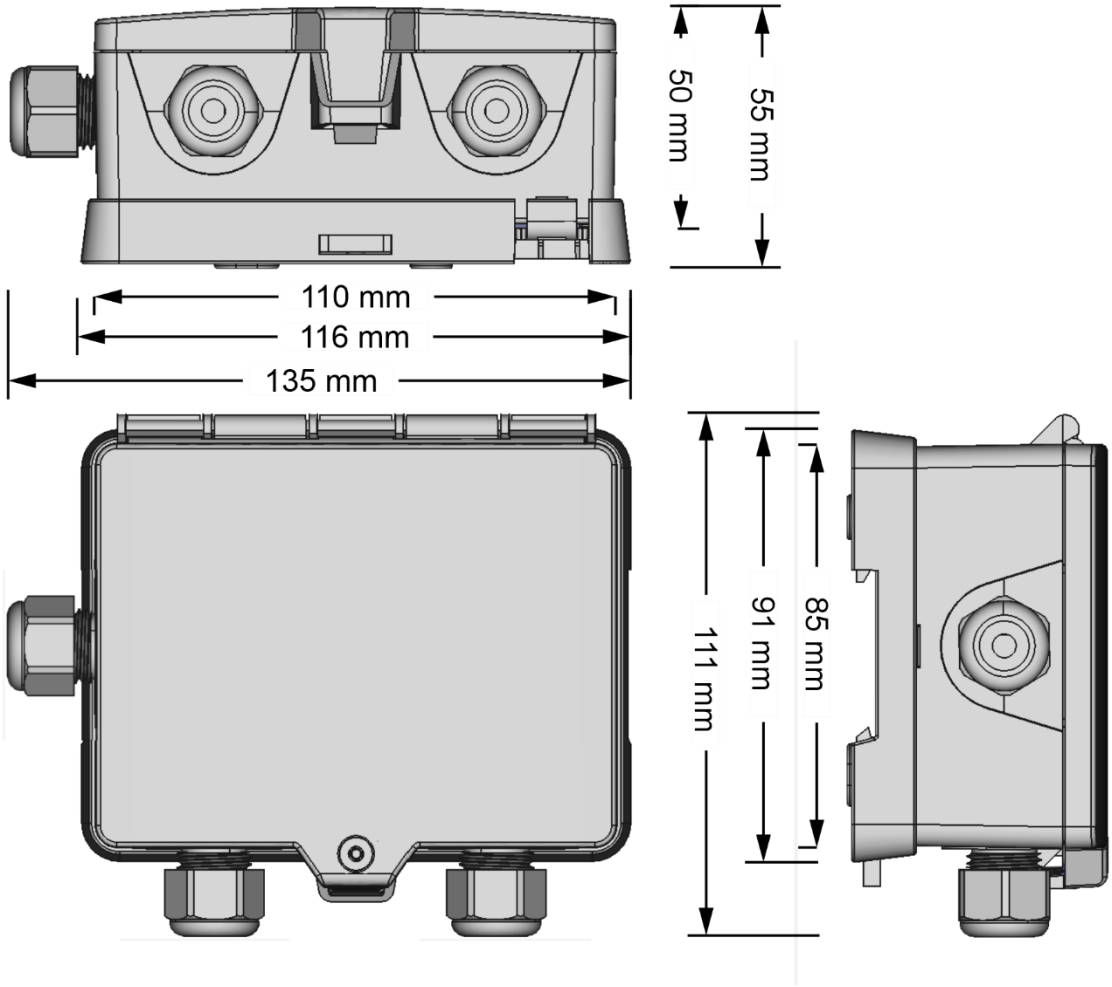
Sensitivity	-137dbm	
Housing		
Assembly	DIN rail & wall	Bracket included in scope of delivery
Material	Polycarbonate	
Protection class	IP65	
Mass	B: 55 mm H: 115 mm - incl. PG screw connection H: 95 mm (mounting surface) L: 115 mm	
Environmental conditions		
6Operating temperature range	-20°C .... +60°C	
Rel. humidity	0 ... 95% (non-condensing)	

8.2 Guidelines and standards

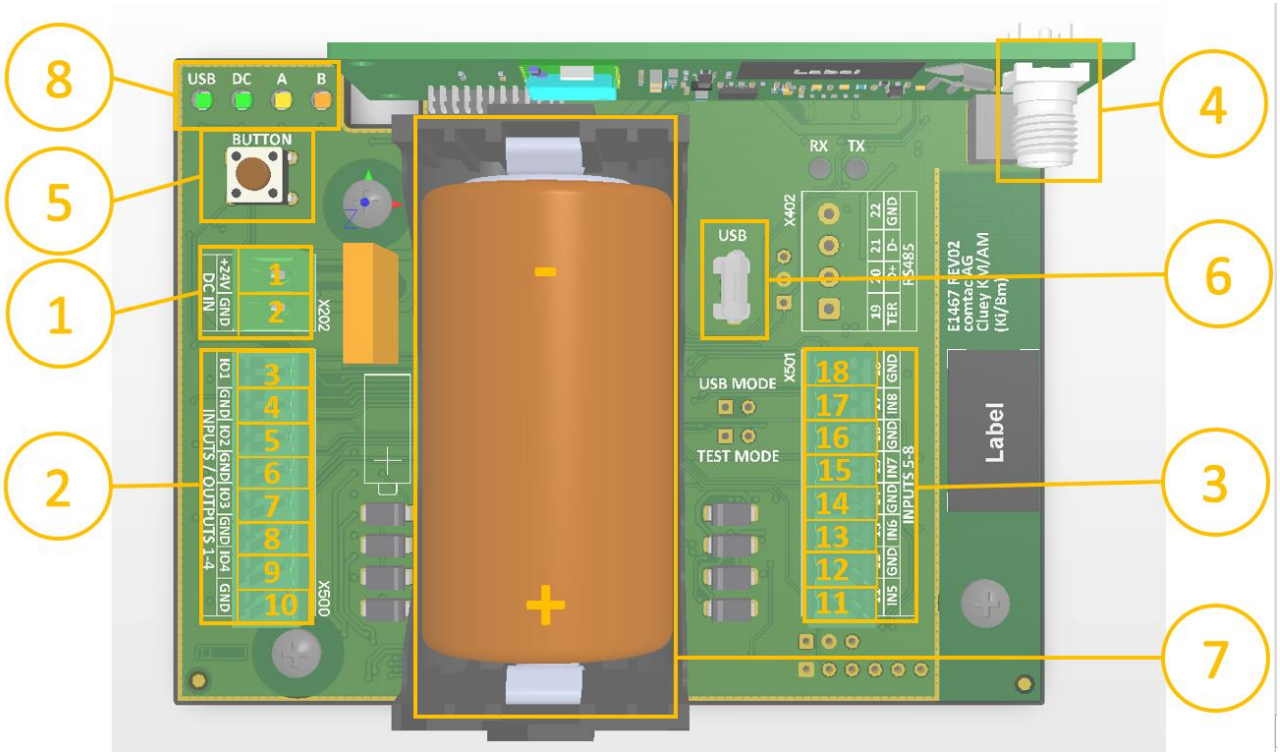
The Cluey is CE compliant. The following directives and standards were applied for the conformity assessment

Directive	Standards	Reference
EMC 2014/30/EU	IEC 61000-6-2:2016 EN IEC 61000-6-2:2019	Interference immunity for industrial areas
	IEC 61000-6-3:2020 EN IEC 61000-6-3:2021	Interference emission for residential, business and commercial areas as well as small businesses
	EN 301 489-3 V2.1.1 (2019-03)	Electromagnetic compatibility and Radio spectrum Matters (ERM) - Electromagnetic Compatibility (EMC) standard for radio equipment and services - Part 3: Specific conditions for short-range devices (SRD) operating on frequencies between 9 kHz and 246 GHz
RED 2014/53/EU	EN 300 220-2 3.2.1 (2018-06)	Short-range devices (SRD) operating in the 25 MHz to 1 000 MHz frequency range
	EN 301 489-3 V2.1.1 (2016-11)	Electromagnetic compatibility and Radio spectrum Matters (ERM) - Electromagnetic Compatibility (EMC) standard for radio equipment and services - Part 3: Specific conditions for Short Range Devices (SRD) operating on frequencies between 9 kHz and 246 GHz
	EN 62479: 2010	Assessment of the compliance of low power electronic and electrical equipment with the basic restrictions for the safety of people in electromagnetic fields (10 MHz to 300 GHz)
RoHS 2011/65/EU	EN 50581:2012	Technical documentation for the assessment of electrical and electronic equipment with regard to the restriction of hazardous substances

8.3 View and dimensions



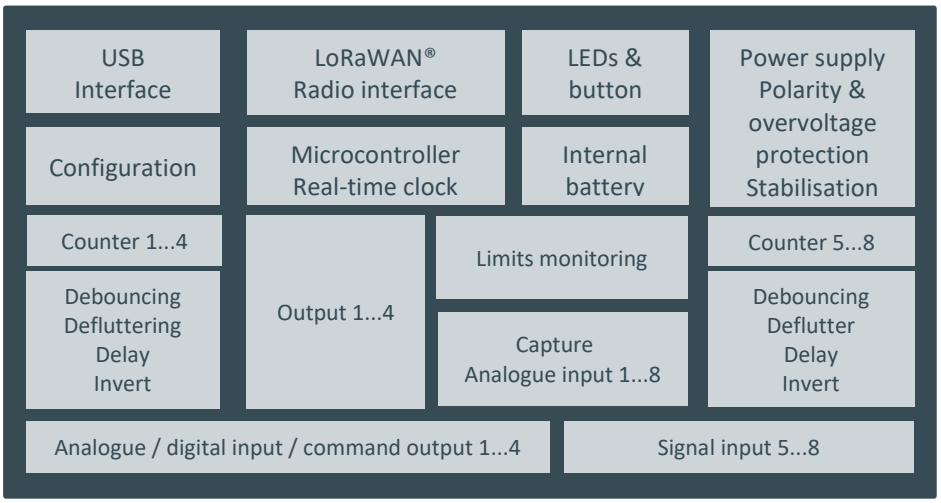
8.4 Connections



Number	Connection designation	PIN / designation	Function
1	DC IN	1	+24V
		2	GND
2	INPUTS/OUTPUT 1-4	3	IO1
		4	GND
		5	IO2
		6	GND
		7	IO3
		8	GND
		9	IO4
		10	GND
3	INPUTS 5-8	11	IN5
		12	GND
		13	IN6
		14	GND
		15	IN7
		16	GND
		17	IN8
		18	GND
4	Antenna		Antenna connection
5	BUTTOM		Button
6	USB		USB configuration interface
7	Battery		Internal battery
8	LEDs	USB IN (green)	USB supply
		DC IN (green)	External power supply
		LED B (yellow)	LED
		LED A (red)	

9 Function

9.1 Function block diagram



9.2 Function overview

The Cluey AM has 8 terminal pairs (IO 1-4 and IN 5-8) whose function can be configured. They can each be used as digital inputs for the acquisition of digital signal states, counting pulse and for operating time acquisition or as analogue inputs for standard current and voltage signals.

The INPUTS/OUTPUT 1-4 connections can also be configured as digital outputs.

9.2.1 Digital inputs

Various configurable processing functions are integrated for the signals acquired via the digital inputs.

Each input is debounced, can be inverted and delayed individually.

The configurable flutter suppression prevents too many telegrams from being sent in the case of "faulty" messages that occur too frequently, thus unnecessarily burdening the battery and radio budget.

Also for optimisation purposes, an adjustable transmission delay is implemented which ensures that in the case of rapidly successive messages at different inputs, only one instead of several successive radio telegrams is sent.

Digital inputs can be configured in pairs for double signal processing. The on/open, off/closed and intermediate positions (both inputs low or high) are then detected and monitored over time. This function is ideally suited for monitoring gates, barriers, sliders, circuit breakers/disconnectors, etc.

In addition, further operating modes - wiper message, pulse counter or operating time counter - are available for each of the digital inputs.

9.2.2 Counter

When configured as a pulse counter, the transmitted count value contains the number of detected pulses.

When configured as an operating time counter (e.g. for runtime monitoring of fans), the counter value is regularly incremented as long as the corresponding input is active. Counter reset and overflow can be detected and distinguished by the status information transmitted with each counter value.

Counters can be reset by downlink command.

9.2.3 Analogue inputs

When configuring the inputs for the acquisition of analogue signals, you can choose between voltage input for 0...10V standard or current input for 0...20mA standard.

The analogue values are monitored for errors, i.e. if the measured value is < 2V or 4mA (wire breakage) or > 11V or 22mA (sensor error), a corresponding measured value status bit is set in each case.

The analogue values can each be additionally monitored for the violation of 2 limit values. The status of the limit value violation is also displayed in status bits (limit bits) belonging to the measured value.

In addition, a delta event function is implemented that enables transmission adapted to the rate of change of the analogue value. The current analogue value is compared with the last transmitted value. If the difference is greater than the configured delta value, a new transmission is triggered.

9.2.4 Digital outputs

The INPUTS/OUTPUT 1-4 connections can also be configured as digital outputs.

The outputs can be switched on or off statically by downlink command or output a pulse with adjustable duration (wiper) by a simple command.

Each output can also be controlled inverted.

The status of the output can be read back via the digital input of the respective connection, the duty cycle can be counted with the operating time counter function or the switching operations can be counted with the pulse counter function.

In addition to control via downlink commands, the outputs can also be controlled directly locally in the unit by the digital inputs or the limit bits of the analogue value limit monitoring.

9.2.5 Supply

Power is supplied via the integrated replaceable battery or external 24V power supply. In battery operation, depending on the configuration, a battery life of more than 10 years can be achieved. The battery's state of charge is transmitted.

9.2.6 Data transmission

The transmission of the input states is event-controlled; the state of the associated inputs is transmitted with the time stamp of the event. In addition, a periodic transmission can be configured.

Data transmission triggering events are:

- Changing the state of the digital input
- Timeout of the intermediate position for double messages
- Activation of the object protection function (walk-in or alarm)
- Blocking or unblocking a digital input using the deflatter function
- Coming or going of the limit value violation for analogue values
- Delta event for analogue values
- Open-circuit or sensor error detection for analogue values
- Failure of the external supply voltage

The respective events can be individually activated or deactivated in the configuration file.

The cause of transmission (COT), event, cyclic or query is also transmitted in the data telegram.

Pulse or operating time counter readings are transmitted periodically at a configurable time, e.g. every Wednesday at 8 p.m.

A downlink command can be used to request transmission of the digital input states, the analogue measured values and the counter readings (general interrogation).

9.2.7 LoRaWAN® communication

The Cluey operates as a Class A device in battery mode, as a Class C device in external power supply mode and in buffer mode in the LoRaWAN® network, so that optimum operation is achieved for the selected function with maximum battery life.

For optimal radio connections, the Cluey has a powerful internal antenna, which can be replaced by a suitable external antenna (not included) in case of difficult radio conditions.

9.2.8 Time

The Cluey has an internal real-time clock for the time stamping of events and the time-synchronous transmission of counter values. To set the real-time clock, the date and time are requested from the network server after the Cluey is started and at configurable intervals via LoRaWAN®.

9.2.9 Configuration

The Cluey is configured by means of a configuration file, which can be accessed via the integrated USB interface.

The configuration parameters can also be read and changed via the LoRaWAN® connection.

9.2.10 Button and LEDs

The status LEDs indicate the status of the supply voltage and the LoRaWAN® connection.


Function of the LEDs

The button can be used to trigger a data transmission, a LoRaWAN® rejoin or a device reset.


Function of the key

9.3 Operating modes

The Cluey can be used for operation

- with external supply voltage with or without battery buffering

and for

- Battery operation can be configured.

The operating mode is set with the parameter


DefaultSupplyMode

9.3.1 Operation with external power supply

If an external supply voltage is connected and the Cluey is configured accordingly, it operates in LoRaWAN® Class C mode, i.e. it can then receive downlink telegrams at any time.

When operating with an external supply, the Cluey can continue to operate from the internal battery in the event of a failure of the external supply, e.g. in order to bridge short supply interruptions and thus also report the failure of the external supply.

Whether and for how long the battery buffered operation is carried out can be set with the parameter

BufferedOperation

In the event of a failure of the external supply, i.e. when switching to battery-buffered operation, the Cluey sends an

Info data packet in its header

the BP: Battery Powered bit in the device status is set accordingly.

After the set bridging time has elapsed parameter

BufferedOperationSpan,

the Cluey switches off and restarts when the external power supply returns.

In the external power supply operating mode, the internal functions as well as the query of the inputs are executed according to the interval set in the configuration (MeasIntervalDcSupply).

The maximum pulse frequency that can be detected without error by the counters and the minimum length for detectable pulses depends on the set cycle time.

9.3.2 Battery operation

In the setting for battery operation (DefaultSupplyMode=1), the Cluey works optimised for low energy consumption to achieve the longest possible battery life.

The Cluey then works as a LoRaWAN® Class A device and polls the inputs according to the - usually longer - cycle time set in the configuration file. All internal functions as well as the query of the inputs take place in the grid of the configured battery operation interval (MeasIntervalBattery).

For long battery life, this cycle time should be chosen as long as possible.

The maximum pulse frequency that can be detected without error by the counters or the minimum pulse length by the digital inputs depends on the set cycle time!

9.4 Inputs

The function of the 8 inputs of the Cluey can be selected by configuration.

The choice is yours:

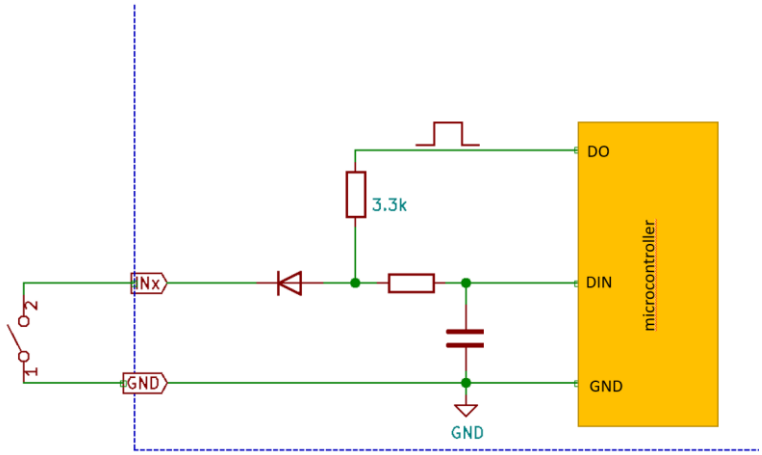
- Digital input
- Analogue input for 0...10V measuring range (voltage input)
- Analogue input for 0...20mA (current input)

9.5 Digital inputs

Two different input configurations are available for the digital inputs, low-power mode and active mode. The low-power mode is optimised for low power consumption, i.e. for battery operation, the active mode has higher switching thresholds and is better adapted to industrial digital signals - as is usual with PLC controls - and is not so well suited for battery operation because of the higher power consumption.

9.5.1 Digital inputs: Low power mode

In Low power mode, the basic input circuit of the digital inputs is as shown below:



The inputs are designed in such a way that the status of a switching contact connected to the input and switching to GND or, for example, an open collector output can be detected.

The switching contact is interrogated cyclically by a short scanning pulse according to the set processing interval.

This method means that the input does not require any additional external switching voltage and that the contact is nevertheless "loaded" with a sufficiently high current* in the closed state so that the closed state can be reliably detected. Due to the fact that the scanning pulses are relatively short, a very small current flows on average when the contact is closed, so that the battery capacity is only slightly loaded.

An open input is processed as active/true/logical 1. An input shorted to GND is processed as inactive/false/logical 0.

The input only works with contacts switching to GND. A contact closing to external positive voltage cannot control the input in this configuration!

This low power mode is always active as long as the active mode is disabled.

Parameter: "IS_Active

*For relays or reed contacts, a minimum current to flow across the contact is necessary for reliable contact making. With contact currents < 1mA, this is usually not guaranteed!

Example Specification of a typical signal relay:

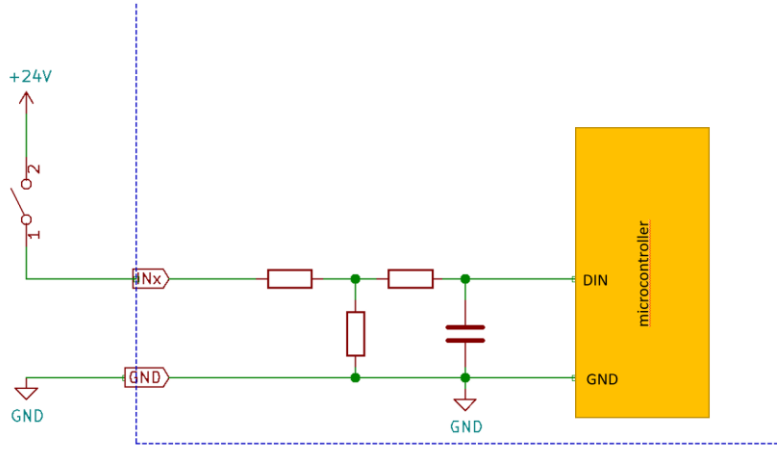
Max. switching power	1250VA / 150W
Min. contact load 1)	No gold plated: 5VDC 10mA Gold plated: 5VDC 1mA
Mechanical endurance	2 x 10^7OPS
Electrical endurance	1 x 10^5OPS (3A 250VAC/30VDC, Resistive load, AgNi, at 85°C, 1s on 9s off) 5 x 10^4OPS (5A 250VAC/30VDC, Resistive load, AgNi, Room temp., 1s on 9s off)

Notes:1) Min. contact load is reference value. Please perform the confirmation test with the actual load before usage since reference value may change according to switching frequencies, environmental conditions and expected life cycles

9.5.2 Digital inputs: active mode

In active mode, digital inputs are designed for control with external voltage. Due to the higher switching thresholds, a higher interference immunity is achieved compared to the low-power inputs and the inputs are compatible with those of PLC systems.

When voltage is applied, a relatively high current flows continuously through the input voltage divider, so this configuration is not well suited for battery operation.



The active mode can be set individually for each input by the

IS_Active

and must be switched on in the configuration file.

An open input or with an input voltage lower than the switching threshold is processed as an inactive/false/logical 0 signal. An input voltage greater than the switching threshold is processed as an active/true/logical 1 signal.

9.5.3 Processing the digital inputs

The inputs can be activated individually in the unit configuration. Only activated inputs are queried and processed with the functions described below and only the activated inputs are considered during data transmission.

Especially in battery operation, only the used inputs should be activated to reduce power consumption. Likewise, the LoRaWAN® telegrams are not unnecessarily prolonged by unused data to extend battery life and data transmission is more optimal in terms of power consumption and duty cycle.

Parameter: "IS_Enable

9.5.3.1 Invert

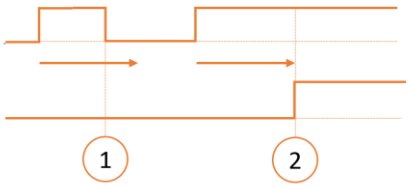
The inputs can be inverted by configuration. This enables the state of the signal to be adapted to the logical function or the function of the connected switching contacts (normally open /normally closed).

Parameter: "IS_Invert

9.5.3.2 Message delay

Inputs can be individually delayed. Delay means that the state of a signal after a change must be stable for the set delay time before the change is accepted.

This can be used, for example, to suppress interference. In addition, in many applications, messages only need to be transmitted to higher-level systems if they are present for a longer period of time.



Input signal

Delayed signal

1: Input signal is active for a shorter time than the set delay time (arrow). Signal change is therefore ignored.

2: Input signal is active longer than the set delay time (arrow): Signal change is accepted.

The function can be activated/deactivated per input.

Parameters: "IS_DelayEnable"; "IS_DelayRising"; "IS_DelayFalling".

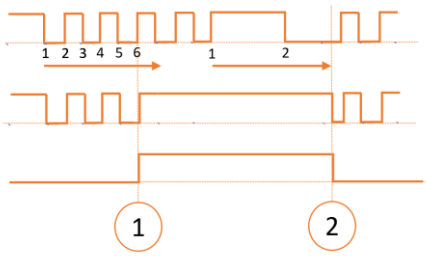
9.5.3.3 Deflutter

The deflutter function is used to suppress inputs that switch too frequently.

If an input changes more frequently than the configured number (IS_DeflutterCount) within the configured monitoring interval (IS_DeflutterInterval), the signal is frozen at the current state and thus does not trigger any further transmission events. The signal then receives the status "blocked".

If the signal subsequently changes less frequently than the configured number within the monitoring interval, the blocked state is cancelled again and changes in the state of the input are accepted and transmitted again.

The deflutter function can be enabled/disabled individually for each input.



Input signal

Fluttered signal

Signal state blocked

1: More than the configured number (here 5) of signal changes occurred during the monitoring interval. The state of the defluttered signal is frozen and the signal state is set to blocked.

2: Less signal changes have occurred within the monitoring time. Blocked state is cancelled and the defluttered signal follows the input signal again.

This function can prevent the transmission of too many unimportant or faulty messages as well as the unnecessary consumption of the transmission time and battery capacity limited by duty cycle regulations.

A typical application for this is the signal from motion detectors.

Parameters: "IS_DeflutterEnable"; "IS_DeflutterInterval"; "IS_DeflutterCount".

9.5.3.4 Double messages

In many applications, e.g. gates, sliders, circuit breakers, ... there is not only the state on/off or open/closed, but in addition to these end states/end positions also an intermediate position for a certain time, during which e.g. a gate opens or closes or a slider runs. For the monitoring of such devices, the Cluey has a double message processing.

Two inputs (IO1 + IO2, IO3+ IO4, IN5+IN6, IN7 + IN8) can be combined in pairs to form a double message. A maximum of 4 double messages can therefore be realised with the 8 inputs of the Cluey.

In contrast to the states of individual inputs, double messages are transmitted with two bits instead of one bit, which represent the state of the double message:

Bit 1	Bit 0	State	Explanation
0	1	1 (OFF)	Bits corresponding to the status of the two inputs
1	0	2 (ON)	Bits corresponding to the status of the two inputs
0	0	0 (intermediate position)	Bits corresponding to the state of the two inputs. Goes over after the monitoring time and fault condition
1	1	3 (fault position)	If the inputs are in a corresponding state or if the intermediate position has been on too long

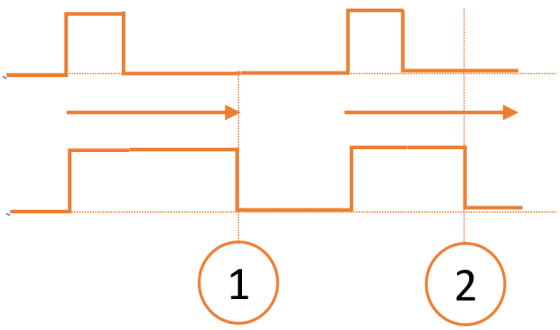
The time monitoring of the intermediate position can be activated or deactivated for each double message and a timeout for the intermediate position can be defined, after which the fault state is displayed.

Parameters: "IS_DoubleIntermediateStateTimoutEnable"; "IS_DoubleIntermediateStateTimout"; "IS_DoubleEnable".

9.5.3.5 Wiper messages

Wiper message processing can be activated individually for each input.

In the process, short pulses are "trapped" at the input, i.e. after the active signal state is held internally until it either receives a corresponding confirmation (at application level) from the higher-level system via downlink, or the configured confirmation timeout (IS_WiperConfirmationTimeout) has expired.



1: Internally held state is set to the current input state because the confirmation timeout has expired

2: A transmission confirmation was received via downlink before the confirmation state expired and the held state was set to the current input state.

The use of this function is particularly useful if only cyclical transmission of the data is configured.

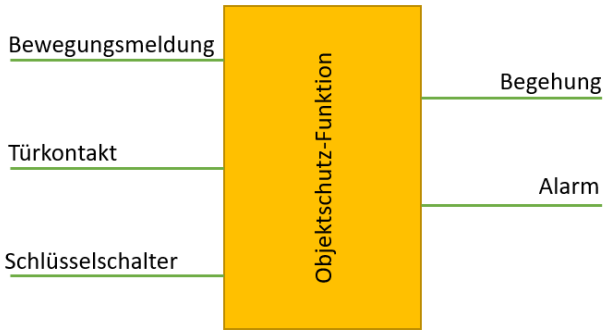
Parameter: "IS_WiperEnable"; "IS_DoubleIntermediateStateTimout"; "IS_WiperConfirmationTimeout".

9.5.3.6 Object protection function / access monitoring

With the object protection function, access monitoring for facilities can be realised. Depending on whether there is authorised or unauthorised access, an access message or an alarm message is triggered.

These walk-in and alarm signals are additional internally generated digital signals to the digital input signals and are considered as "virtual" inputs, transmitted accordingly and can switch outputs of the Cluey locally

The object protection function has 3 logical inputs, each of which can be assigned to one of the 8 digital inputs by configuration.



The three logical inputs of the object protection function are intended for the evaluation of a key switch, a door contact and a motion detector.

The object protection function has two states:

1. Condition "Sharp"

   This state is set as long as the key switch input is inactive. As soon as the motion detector or the door contact become active, the alarm delay is started, after which the alarm message is activated.

By pressing the key switch in time, i.e. before the alarm delay has expired, the alarm message is suppressed and the walk-in message is activated instead.

An active alarm message can be reset by operating the key switch or by resetting via a downlink command.

2. State "Walk-in":

This state is set as soon as the key switch input is active. The walk-in message is then active and the alarm message is inactive. Messages from the motion detector or door contact input are then ignored.

If the key switch contact is deactivated, the walk-in message is deactivated and the alarm delay is started. After the delay has expired, the object protection function returns to the "armed" state and reacts again to door contact and motion detector activations.

The object protection function can be activated and deactivated by configuration.

Parameters: "ACS_Enable"; "ACS_AlarmDelay"; "ACS_MotionDetectorSel"; "ACS_KeySwitchSel"; "ACS_DoorContactSel "

9.5.3.7 Coding of digital values

For data transmission, the digital values are coded as follows. There are three different codings, on the one hand as a single digital value, on the other hand all combined in one data object.

The coding as a single digital value looks like this:

This coding is used in dynamic data packets.

Byte\bit	7	6	5	4	4	2	1	0
1	ObjectType				ObjectNo			
2	COT				Status			
	CYC	INT	EVT	0	0	BL	D1	D0
3	TimeStampOffset MSB							
4	TimeStampOffset LSB							

ObjectType:
- 1: singlePointInfo
- 2: doublePointInfo

ObjectNo:
- No of Input 0...7

COT: Cause of Transmission:
- CYC: Cyclic event
- INT: Interrogation (button, downlink)
- EVT: Event (change of state)

Status:
- BL: Blocked by defluttering function
- D1, D0:

  In case of DoublePointInfo (objectType=2): state of the doublepointinfo Double messages

  In case of singlePointInfo (objectType=1): state of the corresponding digital value

TimeStampOffset: timestamp in seconds in relation to an absolute timestamp, which will be present in datapacket transmitted as well.

The compact coding of all digital values in a data object looks like this:

This coding is used in the static data packet. Single and double messages are not differentiated. The two bits of a double message can be found at the respective bit positions of the associated inputs.

Byte\bit	7	6	5	4	3	2	1	0
1	COT				Status (not used)			
	CYC	INT	EVT	0	0	0	0	0
2	DI 16...9							

3	DI 8...1

For unused digital inputs, the respective bit is always 0

DI8...DI1: State of the digital inputs DI8...DI1

DI16...9: State of the internally formed digital values

- DI9: Walk-in message of the object protection function
- DI10: Alarm message of the object protection function
- DI11...16: reserved for future applications

COT: Cause of Transmission:
- CYC: Cyclic event
- INT: Interrogation (Button, downlink)
- EVT: Event (change of state)

9.5.3.8 Counter

A counter can be activated for each of the 8 digital inputs. The counters can be individually set to work in 2 operating modes, as pulse counters or as operating time counters.

The counters have a length of 24 bits.

Parameters: "IS_CounterEnable"; "IS_CounterMode"; "IS_CounterScaling".

9.5.3.8.1 Pulse counter

As a pulse counter, the pulses detected at the corresponding input are counted. The counter counts on the rising or falling edge of the input, depending on whether the input is configured as inverting or non-inverting.

9.5.3.8.2 Operating time counter

As operating time counters, the counters count the milliseconds, seconds, minutes or hours that the associated input is active, depending on the configuration. The activated inversion of the input is considered.

The counter readings are not retained in the event of a power interruption and are reset to 0. The counters can also be reset by a downlink command. A reset by power interruption or by command is indicated by a set reset status bit(RES) assigned to the counter value.

When a counter overflows for the first time, its overflow status bit (OV) is set. In this way, the reset and overflow bit of a counter can be used to clearly distinguish between overflow and reset (and processed accordingly in the higher-level system).

9.5.3.9 Counts Coding

Counter values are coded as follows during data transmission:

For count values:

Byte\bit	7	6	5	4	4	2	1	0
1	ObjectType				ObjectNo			
	0x41...0x48							
2	COT				Status			
	CYC	INT	EVT	LIM	0	0	RES	OV
3	Count MSB							
4	Count							
5	Count LSB							

ObjectType: 4 = Count value

ObjectNo: Number of the input 1...8

COT: Cause of transmission
- CYC: Cyclic event
- INT: Interrogation triggered by button, downlink command or first transmission after restart or rejoin

- EVT: not used
- LIM: not used

Status:

- OV: Meter overflow
- RES: Counter reset (since last transmission)

9.6 Analogue inputs

The 8 signal connections can be configured as analogue inputs, either as voltage inputs for 0...10V signals or as current inputs for 0...20mA signals.

The function of the inputs is set with the

Parameter: "AIS_Enable" set.

If a signal connection is configured as an analogue input, it is no longer available as a digital input, even if it is enabled as a digital input in the parameterisation.

The inputs are scanned at cyclic configurable intervals.

Parameter: "MeasIntervalAi

The recorded analogue values can be monitored for limit value violation. Two limit values are available for each analogue input. In addition, an overrange or an invalid measured value that may be too small is signalled. This can be used for sensor error and wire break detection.

The analogue signals are transmitted cyclically and/or event-triggered in the event of a limit value violation, overrange or invalid measured value. Delta event triggered transmission can also be set.

Regardless of whether a 0...10V voltage input or a 0...20mA current input is configured, the measured values are always scaled internally to the value range 0...10000:

Example values are:

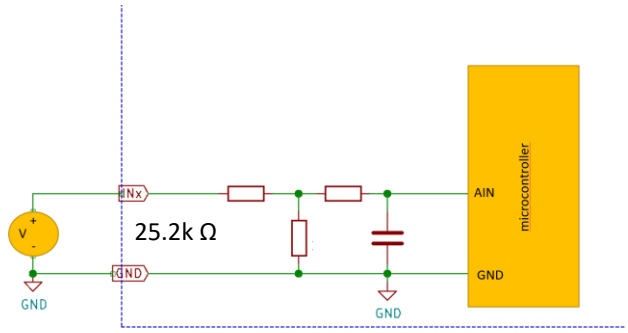
Input current	Input voltage	Internal and transmitted value	Comment
0mA	0V	0	
3.6mA	1.8V	1800	Limit value for wire break detection, invalid analogue value
4mA	2V	2000	
10mA	5V	5000	
20mA	10V	10000	
21mA	10.5V	10500	Limit value for sensor error, overrange
25mA	12.5V	12500	

Configurable values, such as limit values or values for delta event, always refer to the internal measured value representation, i.e. 0...10000.

A differentiation between 0...20mA and 4...20mA (or 0...10V and 2...10V) sensor signals is not provided. The input current or the input voltage is always measured and scaled to the value range 0...10000 without deducting any offset.

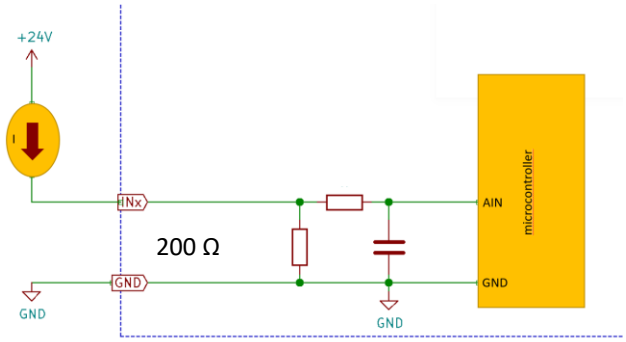
9.6.1 0...10V voltage input

When configuring an input as a voltage input, the equivalent input circuit looks like the following:



9.6.2 **0...20mA current input**

When configured as a current input, the equivalent input circuit looks as follows:



It must be ensured that no impermissibly high voltage can occur at the input (e.g. accidental connection of the supply voltage), this can lead to overloading and destruction of the internal shunt resistor.

9.6.3 **Analogue value processing**

The analogue values acquired via the inputs can be further processed with the following functions.

9.6.4 **Limit value monitoring**

Two limit values are available for each of the maximum 8 analogue values. The currently recorded value is compared with the two limit values.

The limit values are set with the parameters:

AIS_Limit1Value, AIS_Limit2Value

The limit value processing distinguishes between upper and lower limit values. For both limit values the mode with the parameter

AIS_Limit1Direction, AIS_Limit1Direction

can be set to "upper " or "lower ".

Each limit value also has a hysteresis, the size of which can be set with the parameters:

AIS_Limit1Hysteresis, AIS_Limit2Hysteresis

Furthermore, the activation or deactivation of a limit value violation can be delayed. The parameters

AIS_Limit1Enable, AIS_Limit2Enable

for activating or deactivating the delay,

AIS_Limit1DelayRising, AIS_Limit2DelayRising, AIS_Limit1DelayFalling, AIS_Limit2DelayFalling

to set the delay time for incoming or outgoing limit value violations and

AIS_LimitDelayScaling

for scaling the delay time. The scaling allows the delay times to be set in a wide range.

Limit value violations are signalled internally with bits "LIMIT1" (LIM1) and "LIMIT2" (LIM2) per analogue value.

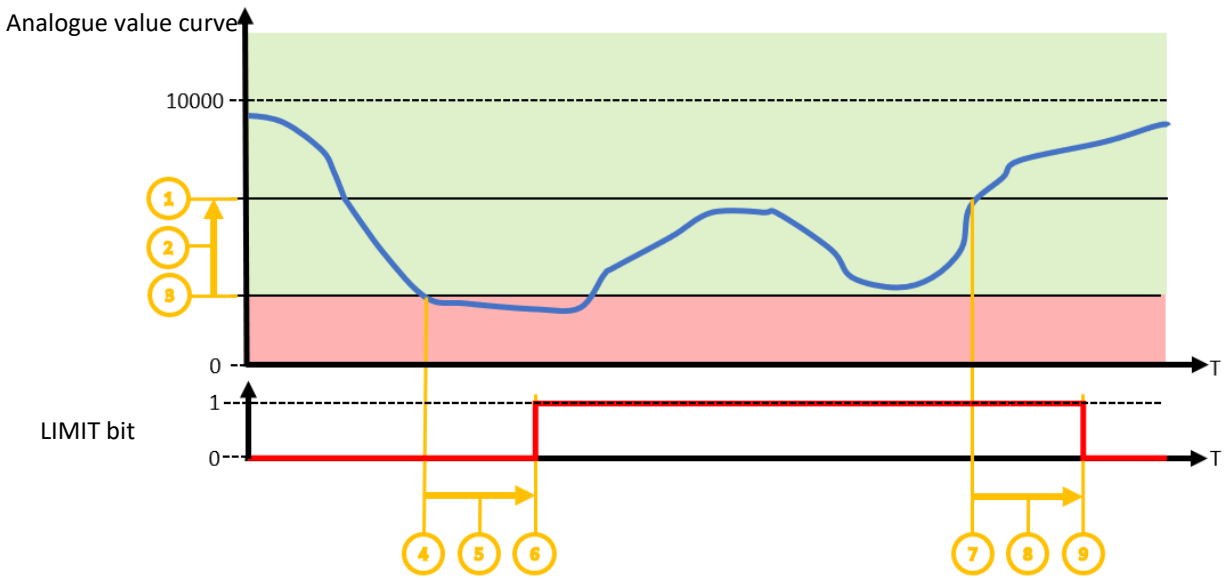
These limit bits are also transmitted during the transmission of the analogue values in the status of the analogue value and can trigger a transmission event when changed as well as being used to control the digital outputs.

9.6.5 Limit value mode "lower

When the limit value mode is set to "lower", a limit value violation is active, i.e. the associated "limit" bit is set, if the analogue value is **less than or equal to the** set limit value. The limit value violation becomes inactive, and thus the limit bit is reset, when the current analogue value is **greater than (limit values + hysteresis).**

The limit bits are only set if the limit value has been active for longer than the delay time "AIS_Limit1DelayRising" or "AIS_Limit2DelayRising" and only reset if the value has fallen below the limit value (limit hysteresis) for the delay time "AIS_Limit1DelayFalling" or "AIS_Limit2DelayFalling".

The function is shown schematically in the following diagram:



(1) Limit value + hysteresis: Value at which the limit value violation is reset if the current analogue value is larger

(2) Hysteresis

(3) Limit value: Value at which the limit value violation is set when the current analogue value is smaller

(4) Upcoming limit value violation: "Rising" delay time is started

(5) Delay time for coming limit violation: AIS_Limit1DelayRising resp.

(6) Delay time expired, limit bits "LIMIT1" or "LIMIT2" are set.

(7) On-going limit value violation: "Rising" delay time is started

(8) Delay time for outgoing limit violation: "AIS_Limit1DelayFalling" or "AIS_Limit1DelayFalling".

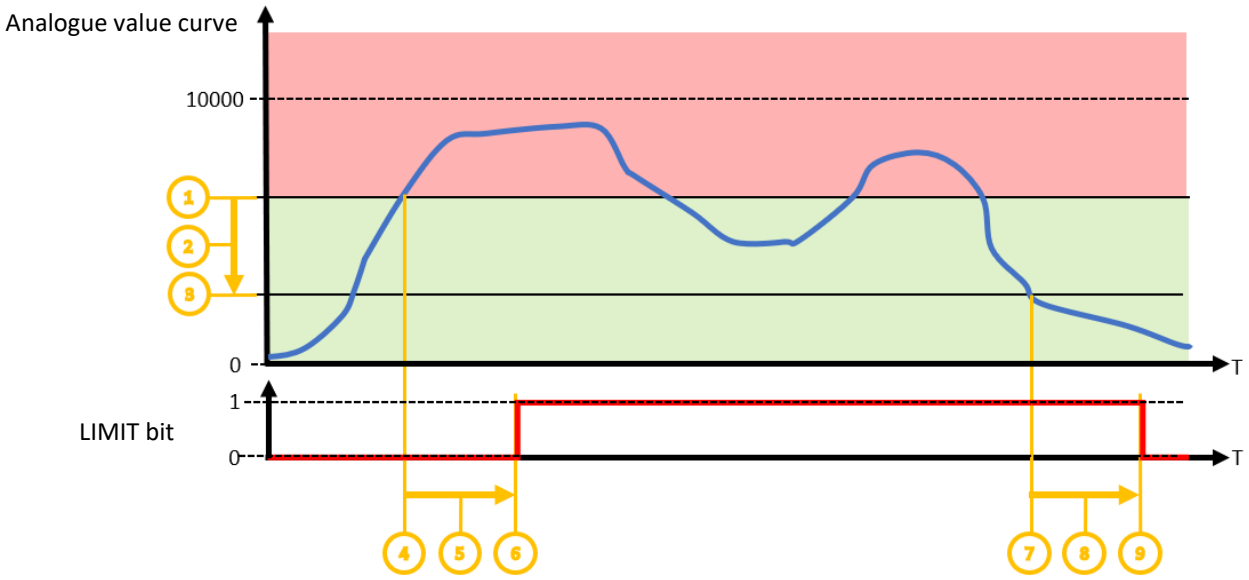
(9) Delay time expired, limit bits "LIMIT1" or "LIMIT2" are set.

9.6.6 Limit mode "upper

When the limit value mode is set to "upper", a limit value violation is active, i.e. the associated "limit" bit is set, if the analogue value is **greater than or equal to the** set limit value. The limit violation becomes inactive, and thus the limit bit is reset, when the current analogue value is **less** than **(limit values -hysteresis**).

The limit bits are only set if the limit value has been active for longer than the delay time "AIS_Limit1DelayRising" or "AIS_Limit2DelayRising" and only reset if the value has fallen below the limit value (limit hysteresis) for the delay time "AIS_Limit1DelayFalling" or "AIS_Limit2DelayFalling".

The function is shown schematically in the following diagram:



(1) Limit value: Value at which the limit value violation is set if the current analogue value is larger

(2) Hysteresis

(3) Limit value hysteresis: Value at which the limit value violation is reset if the current analogue value is smaller.

(4) Upcoming limit value violation: "Rising" delay time is started

(5) Delay time for coming limit violation: AIS_Limit1DelayRising resp.

(6) Delay time expired, limit bits "LIMIT1" or "LIMIT2" are set.

(7) On-going limit value violation: "Rising" delay time is started

(8) Delay time for outgoing limit violation: "AIS_Limit1DelayFalling" or "AIS_Limit1DelayFalling".

(9) Delay time expired, limit bits "LIMIT1" or "LIMIT2" are set.

9.6.7 **Delta Event**

The delta event function is used for dynamic transmission of analogue values. A transmission event is triggered when the analogue value has changed by a certain amount compared to the last transmitted value. This means that I am transmitted less frequently when the analogue value changes slowly and more frequently when the analogue value changes quickly.

For the configuration of the function, the parameters

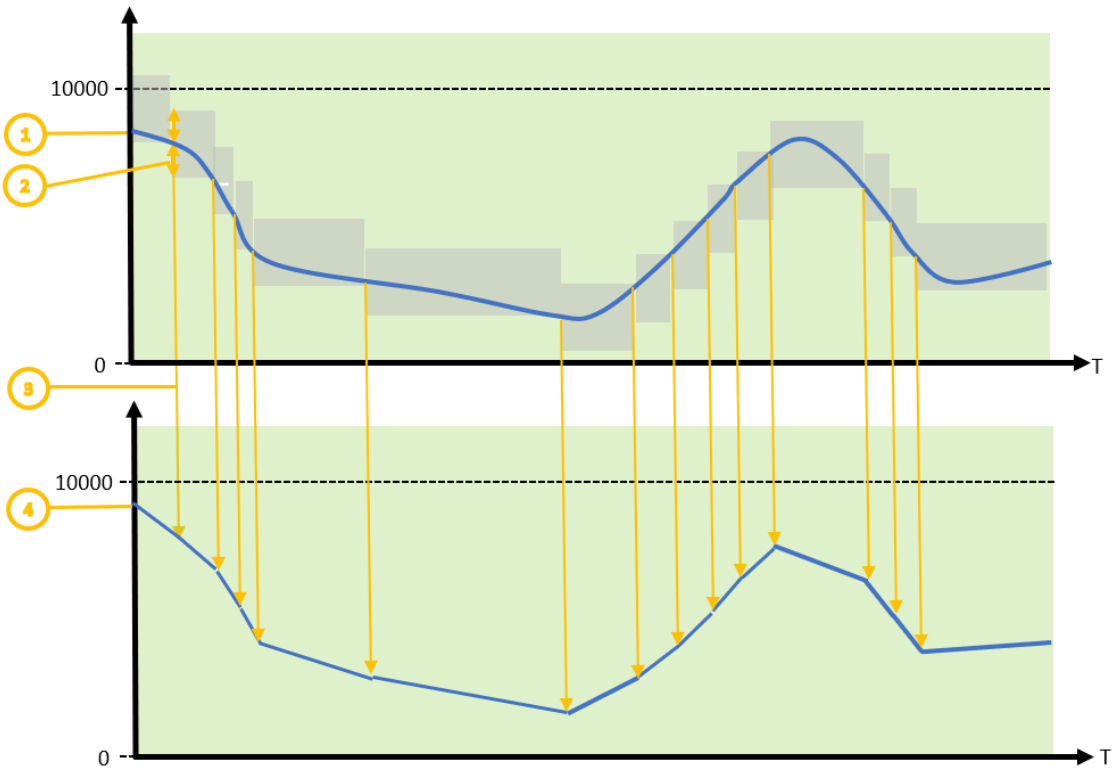
AIS_DeltaEnable

with which the function can be individually activated or deactivated for each analogue value and

AIS_DeltaValue

for setting the tolerance, also individually for each analogue value, is available.

The function is shown in the following diagram:



(1) Analogue value curve
(2) Tolerance range: set by parameter "AIS_DeltaValue", if the current analogue value leaves this range, a delta event is triggered for transmission
(3) Delta Events
(4) Analogue value curve reconstructed in the superordinate system

The actual analogue value curve (1) is shown above. When the measured value leaves the grey area (2), a delta event (3) is triggered and the current analogue value is transmitted.
The analogue value curve (4) in the receiving higher-level system therefore follows (with linear interpolation between the transmitted values) the actual measured value curve with a deviation corresponding to the configured tolerance "AIS_DeltaValue".
The transmission frequency therefore depends on the rate of change of the analogue value.

9.6.8 Overrange/sensor error detection

The recorded analogue values are checked for exceeding the additional fixed limit value:

Input current	Input voltage	Internal fixed limit Overrange/sensor error detection
21mA	10.5V	10500

Exceeding this limit value can be interpreted as the sensor exceeding the measuring range or as the sensor signalling that there is an error (according to Namur guideline).

The change of this limit value state can trigger a transmission if this event is activated.

The limit value status is transmitted in the status information of the respective analogue value as "OVERFLOW" bit (OV).

The function is shown schematically in the following section.

9.6.9 Underrange/sensor wire break detection

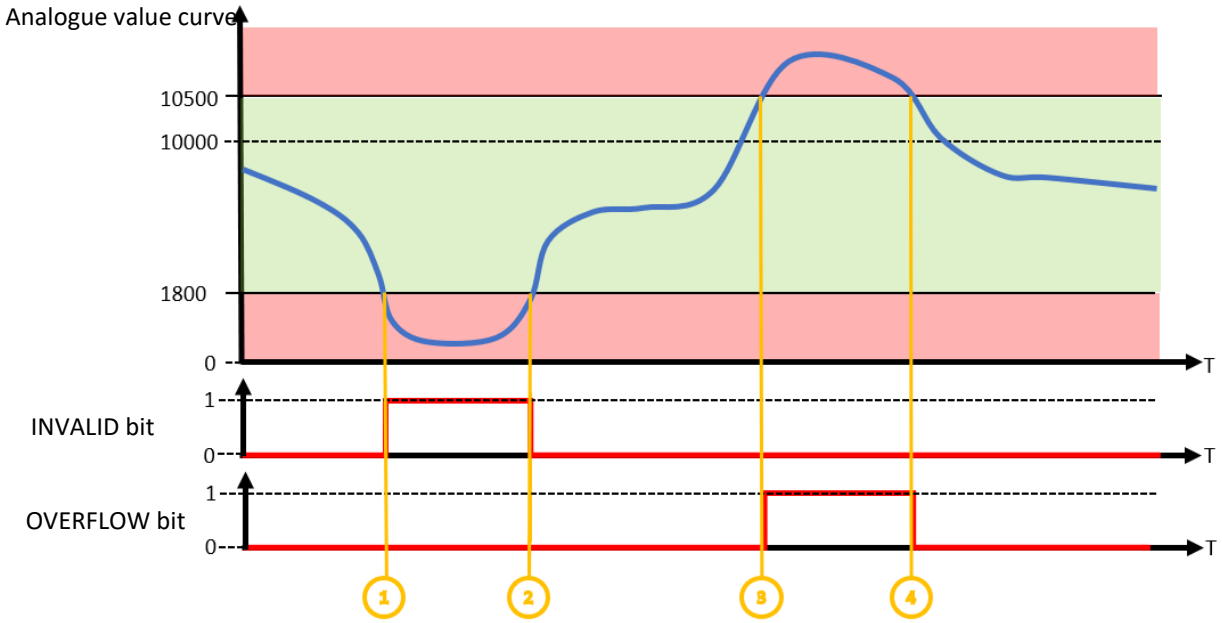
The recorded analogue values are also checked for falling below another fixed limit value:

Input current	Input voltage	Internal fixed limit value for Underrange/sensor wire break detection
3.6mA	1.8V	1800

When using 4...20mA or 2...10V sensors, falling below this limit value can be interpreted as falling below the measuring range or as an indication of a wire break in the sensor cable.

The change of this limit value state can trigger a transmission if this event is activated.

The limit value status is transmitted in the status information of the respective analogue value as "INVALID" bit (IV).



(1) Underrange/sensor wire break detection limit, INVALID bit is set

(2) Exceeding the limit value for underrange/sensor wire break detection, INVALID bit is reset

(3) Exceeding of the limit value for overrange/sensor error detection, OVERFLOW bit is set.

(4) Undercutting of the limit value for overrange/sensor error detection, OVERFLOW bit is reset

9.6.10 **Analogue value coding**

During data transmission, analogue values are coded as follows.

Byte\bit	7	6	5	4	4	2	1	0
1	ObjectType				ObjectNo			
	0x51 to 0x58							
2	COT				Status			
	CYC	INT	EVT	LIM	IV	OV	LIM2	LIM2
3*	0							
4 / 3	Analogue value MSB							
5 /4	Analogue value LSB							

ObjectType: 5 = Analogue value

ObjectNo: Number of the input 1...8

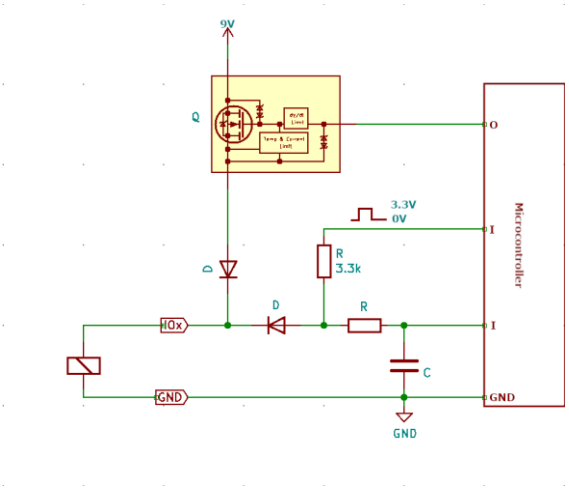
COT: Cause of transmission
- CYC: Cyclic event
- INT: Interrogation triggered by button, downlink, or at first transmission after restart or rejoin
- EVT: Delta Event
- LIM: Limit value/measuring range overrange/underrange

Status:
- IV: invalid, below measuring range
- OV: overflow, measuring range exceeded
- LIM1: Limit value1 violated
- LIM2: Limit value2 violated

(*) : Byte 3 is only present if static payload_format has been selected in the configuration.

9.7 Digital outputs

The Cluey has 4 outputs that share the connection terminals (IO1, IO2, IO3, IO4) with the corresponding inputs.



The outputs can be enabled/disabled individually by configuration. If an output is disabled, control by downlink command or due to internal linking is ignored.

The outputs, like the inputs, can be inverted and configured for static or wiper operation.

The outputs can be controlled by downlink commands and/or controlled locally by digital inputs or limit bits.

In static mode, the output is set according to the target state (on/off) transmitted in the downlink command, considering any configured inversion.

In wiper mode, the output is activated for the configured wiper time after receiving a corresponding downlink command and then automatically deactivated again. If the downlink command contains a wiper time, this time is used instead of the configured time.

Parameters: "OS_Enable"; "OS_Invert"; "OS_Mode"; "OS_WiperTime

The inputs parallel to each output can be used simultaneously. For example, to read back the status of the outputs or to record the duty cycle or the number of switching operations of the digital input with the counting function.

9.8 Trigger function

The Cluey has a periodic trigger function. This function can be used to switch on the digital outputs cyclically.

This allows, for example, an external sensor to be switched on periodically for measurements so that it operates in interval mode. This is particularly interesting for battery- or solar-powered systems in order to greatly reduce the average power consumption.

For controlling the outputs, the trigger is available as an internally formed digital message (bit 11).

The trigger period can be set in a wide range with the parameters

OS_TriggerPeriod , OS_TriggerPeriodScaling

between seconds and several years.

The function can be started with the parameter

OS_TriggerEnable

can be activated or deactivated.

9.9 Local control of the outputs

An output can be controlled locally by each digital input. It can be specified whether the output

- is **set** by a change of an input **(**i.e. a change from 0→ 1),
- or is **reset** a change of an input an input **(**0→ 1)**,**
- or whether the output **follows** an input**,** i.e. is set on rising input signal (0→ 1) and reset on descending input signal (1→ 0).

To reverse the logic, the input or the output can be inverted. If the input is inverted, the output is controlled with the falling edge at the input (0→ 1). If the output is inverted, the setting and resetting of the output are reversed and the "Following" is also inverted accordingly.

In addition, each limit bit (LIMIT) of the analogue inputs and the alarm and access message (ACS) generated by the object protection function can control the outputs locally in the same way as digital inputs.

An output configured as a wiper can only be set, not reset, by a digital input or the other messages. Resetting is always done automatically after the wiper time set in the configuration has elapsed.

It should be noted that the control is purely edge-controlled. This allows an output to also be controlled by several digital inputs, limit bits and by downlink command.

The configuration of the local control of the outputs 1,2,3,4 by the inputs or internally formed digital values is done via the parameters

"IOMS_Out1", "IOMS_Out2", "IOMS_Out3", "IOMS_Out4"

The assignment of the digital values is as follows:

	Digital input IO1	Digital input IO2	Digital input IO3	Digital input IO4	Digital input IN5	Digital input IN6	Digital input IN7	Digital input IN8	Object protection inspection	Object protection alarm	reserved	reserved	reserved	reserved	reserved	reserved
	DI	_	_	_	_	_	_	_	ACS	_	_	_	_	_	_	_
IOMS_Outx=	0;	0;	0;	0;	0;	0;	0;	0;	0;	0;	0;	0;	0;	0;	0;	0;

The configuration of the local control of the outputs 1,2,3,4 by the limit value bits is done via the parameters

"LOMS_Out1", "LOMS_Out2", "LOMS_Out3", "LOMS_Out4"

Both sets of parameters each form an assignment matrix that allows each output to be assigned to each digital input and each limit value bit for control.

The allocation of the limit values is as follows:

	Limit value 1 Analogue input IO1	Limit value 1 Analogue input IO2	Limit value 1 Analogue input IO3	Limit value 1 Analogue input IO4	Limit value 1 Analogue input IN5	Limit value 1 Analogue input IIN6	Limit value 1 Analogue input IN7	Limit value 1 Analogue input IN8	Limit value 2 Analogue input IO1	Limit value 2 Analogue input IO2	Limit value 2 Analogue input IO3	Limit value 2 Analogue input IO4	Limit value 2 Analogue input IN5	Limit value 2 Analogue input IIN6	Limit value 2 Analogue input IN7	Limit value 2 Analogue input IN8
	LIMIT1	_	_	_	_	_	_	_	LIMIT2	_	_	_	_	_	_	_

LOMS_Outx=	0;	0;	0;	0;	0;	0;	0;	0;	0;	0;	0;	0;	0;	0;	0;	0;

9.9.1 Example 1: Local control of an output by different inputs

Output **1 is** to be switched on by (e.g. a push button on) input 5 (function "set"=>1) and switched off by input 6 (function "reset"=>2).

Parameter setting:

```
Output Settings (OS):
OS_Enable=1;0;0;0 (0: not enabled, 1: enabled)
OS_Invert=0;0;0;0 (0: not inverted 1: inverted)
OS_Mode=0;0;0;0 (0: static, 1: wiper)
OS_WiperTime=00150;00150;00150;00150 (0..65535 ms)


IO Mapping Settings (IOMS):
_________|DI_____________|ACS____________
IOMS_Out1=0;0;0;0;1;2;0;0;0;0;0;0;0;0 (Output control by DI: 0: not enabled, 1: set, 2: reset , 3: follow)
...
```

9.9.2 Example 2: Local control of an output by limit values

Output 2 is to issue a pulse command (e.g. slider open) when the limit value 1 of analogue input 5 is undershot and output 2 is to issue a pulse command (e.g. slider closed) as soon as the limit value 2 of analogue input 5 is exceeded.

- Input IN5 is configured as an analogue input (current input)
- Limit value 1 of analogue input 5 is configured as a lower limit value (20%). Then limit value bit 1 is set when the value falls below the set limit value.
- Limit value 2 of analogue input 5 is configured as an upper limit value (80%). Then limit value bit 2 is set when the set limit value is exceeded.
- IO connections 1 and 2 are each configured as output and for wipe pulse output with wipe duration 5 seconds
- Assignment of the inputs to the outputs.

```
Analogue Input Settings (AIS):
AIS_Enable=0;0;0;0;2;0;0 (0: not enabled, 1: 0-10V, 2: 0-20mA)
AIS_DeltaEnable=0;0;0;0;0;0;0 (0: not enabled, 1: enabled)
AIS_DeltaValue=00500;00500;00500;00500;00500;00500;00500;00500 (0..65535)
AIS_Limit1Enable=0;0;0;1;0;0;0 (0: not enabled, 1: enabled)
AIS_Limit2Enable=0;0;0;1;0;0;0 (0: not enabled, 1: enabled)
AIS_Limit1DelayEnable=0;0;0;0;0;0;0 (0: not enabled, 1: enabled)
AIS_Limit2DelayEnable=0;0;0;0;0;0;0 (0: not enabled, 1: enabled)
AIS_LimitDelayScaling=0;0;0;0;0;0;0 (0: ms, 1: s, 2: min, 3: h)
AIS_Limit1DelayRising=00100;00100;00100;00100;00100;00100;00100;00100 (0..65535)
AIS_Limit2DelayRising=00100;00100;00100;00100;00100;00100;00100;00100 (0..65535)
AIS_Limit1DelayFalling=00100;00100;00100;00100;00100;00100;00100;00100 (0..65535)
AIS_Limit2DelayFalling=00100;00100;00100;00100;00100;00100;00100;00100 (0..65535)
AIS_Limit1Value=08000;08000;08000;08000;02000;08000;08000;08000 (0..65535)
AIS_Limit2Value=08000;08000;08000;08000;08000;08000;08000;08000 (0..65535)
AIS_Limit1Hysteresis=00500;00500;00500;00500;00500;00500;00500;00500 (0..65535)
AIS_Limit2Hysteresis=00500;00500;00500;00500;00500;00500;00500;00500 (0..65535)
```

```
AIS_Limit1Direction=0;0;0;0;0;0 (0: lower, 1: upper)
AIS_Limit2Direction=0;0;0;1;0;0 (0: lower, 1: upper)
.......
Output Settings (OS):
OS_Enable=1;1;0;0 (0: not enabled, 1: enabled)
OS_Invert=0;0;0;0 (0: not inverted 1: inverted)
OS_Mode=1;1;0;0 (0: static, 1: wiper)
OS_WiperTime=05000;05000;00150;00150 (0..65535 ms)
....
Limit-Output Mapping Settings (LOMS):
_________|LIMIT1_________|LIMIT2_________
LOMS_Out1=1;0;0;0;0;0;0;0;0;0;0;0 (Output control by Limit: 0: not enabled, 1: set, 2: reset, 3: follow)
LOMS_Out2=0;0;0;0;0;0;1;0;0;0;0;0 (Output control by Limit: 0: not enabled, 1: set, 2: reset, 3: follow)
LOMS_Out3=0;0;0;0;0;0;0;0;0;0;0;0 (Output control by Limit: 0: not enabled, 1: set, 2: reset, 3: follow)
LOMS_Out4=0;0;0;0;0;0;0;0;0;0;0;0 (Output control by Limit: 0: not enabled, 1: set, 2: reset, 3: follow)
```

9.9.3 Example 3: Local control of an output by an input

Output **3** is to indicate the state of input 7 inverted. For example, if there is no voltage at input 7, a lamp should be switched on at the output.

- Configure connection IO3 as a static output
- Connection 7 is configured as a digital input
- Assignment of input 7 to output 3 so that output follows input
- Inverting the output 1, so that 0→ 1 change at the input causes a switch-off and a change 1→ 0 causes a switch-off of the output

```
....
Output Settings (OS):
OS_Enable=0;0;1;0 (0: not enabled, 1: enabled)
OS_Invert=1;0;0;0 (0: not inverted 1: inverted)
OS_Mode=0;0;0;0 (0: static, 1: wiper)
OS_WiperTime=05000;05000;05000;05000 (0..65535 ms)
....

IO Mapping Settings (IOMS):
_________|DI_____________|ACS____________
IOMS_Out1=0;0;0;0;0;0;0;0;1;0;0;0;0;0 (Output control by DI: 0: not enabled, 1: set, 2: reset , 3: follow)
IOMS_Out2=0;0;0;0;0;0;0;0;0;0;0;0;0;0 (Output control by DI: 0: not enabled, 1: set, 2: reset , 3: follow)
IOMS_Out3=0;0;0;0;0;0;3;0;0;0;0;0;0;0 (Output control by DI: 0: not enabled, 1: set, 2: reset , 3: follow)
IOMS_Out4=0;0;0;0;0;0;0;0;0;0;0;0;0;0 (Output control by DI: 0: not enabled, 1: set, 2: reset , 3: follow)
```

9.9.4 Example 4: Local control of an output by periodic trigger

Output **1 can** be switched on for 10 seconds every hour.

- Configure connection IO1 as wiper output,
- Activate trigger function and set trigger period and scaling for 1 hour period duration
- Assignment of the trigger to control output 1, so that output is set by trigger

```
OS_Enable=1;0;0;0 (0: not enabled, 1: enabled)
OS_Invert=0;0;0;0 (0: not inverted 1: inverted)
OS_Mode=1;0;0;0 (0: static, 1: wiper)
OS_WiperTime=10000;05000;05000;05000 (0..65535 ms)
OS_TriggerEnable=1 (0: not enabled, 1: enabled)
OS_TriggerPeriod=60 (1..65535 s)
OS_TriggerPeriodScaling=1 (0: s, 1: min, 2: h)

IO Mapping Settings (IOMS):
_________|DI_____________|ACS___________
IOMS_Out1=0;0;0;0;0;0;0;0;1;0;0;0;0;0 (Output control by DI: 0: not enabled, 1: set, 2: reset , 3: follow)
IOMS_Out2=0;0;0;0;0;0;0;0;0;0;0;0;0;0 (Output control by DI: 0: not enabled, 1: set, 2: reset , 3: follow)
IOMS_Out3=0;0;0;0;0;0;0;0;0;0;0;0;0;0 (Output control by DI: 0: not enabled, 1: set, 2: reset , 3: follow)
IOMS_Out4=0;0;0;0;0;0;0;0;0;0;0;0;0;0 (Output control by DI: 0: not enabled, 1: set, 2: reset , 3: follow)
```

9.10 Function of the LEDs

The Cluey has 4 LEDs to signal certain states. Two green LEDs, "USB" and "DC IN", signal the current supply voltage, i.e. either an external DC voltage source or a plugged-in USB cable. Neither of the two LEDs is lit when the unit is powered by batteries.

During operation, the yellow LED "A" serves to signal a LoRaWAN® communication. In the case of an active transmission, this LED therefore flashes.

If the orange LED "B" lights up briefly at the end of a transmission, it signals an error in the transmission or an unsuccessful join attempt.

Since LEDs naturally have a non-negligible power consumption, they are not active in battery mode. In battery mode, the status LEDs are only used during the first LoRaWAN® transmissions (during time synchronisation and the first measured value transmission). After that, the status LEDs are permanently switched off. To help with troubleshooting, the status LEDs are switched on again during a transmission of the input and counter status that was triggered by a button press.

During operation with an external power supply, the status LEDs are permanently on.

Summary:

Name	Colour	Function
USB IN	Green	USB cable connected. The unit is either in the bootloader or the configuration file is provided
DC IN	Green	External DC power supply available
LED A*	Yellow	Signalling a LoRa transmission
LED B*	Red	Error during a LoRa transmission or unsuccessful join attempt

*Does the Cluey find itself in the

 Firmware update mode

LED "A" and LED "B" flash alternately.

9.11 Function of the key

The "Button" key can be used to trigger different functions depending on the length of the keystroke:

Keystroke duration	Function
<2s	Triggering a LoRa transmission If (with OTAA) a join request has not yet been sent, a join request is sent.
>5s	Sending a Join Request
>10s	Reset of the Cluey

10 Antenna connection

The antenna connection has an impedance of 50 Ohm in the frequency range 868MHZ.

Only suitable internal or external antennas may be connected. Incorrect matching, especially operation without an antenna or with an unsuitable antenna, leads to a reduction in performance and can damage the unit.

For antennas connected by longer antenna cables, we recommend using low-attenuation antenna cables.

11 USB port

The USB port is used to configure the unit. It is designed as a USB-C connection. When connected to a PC or similar, the unit appears as a USB drive on which a configuration file "cfg.txt" is located.

See section: Configuration

As a rule, no special drivers are required for the connection. They are included in the common operating systems.

11.1 Data transmission, events

The transmission of the recorded data is event-driven. For this purpose, the recorded states are checked for changes, for example. If there is a change, the time is recorded and written together with the current value of the signal as a data object in the buffer for data to be sent.

11.1.1 Events

The following events are defined in the Cluey. They can each be enabled/disabled, which determines whether the respective event triggers a transmission of the data object.

A transmission can also be triggered by pressing and by downlink commands→ General interrogation.

11.1.1.1 Events for digital inputs

A transmission can be triggered with the help of the events of the digital signals when the value changes, the status changes and/or cyclically. This can be activated or deactivated for the various events with the corresponding parameter.

Event	Parameters for enable/disable	Description
Upcoming message	ES_RisingEnable	Triggered when the signal changes to the active state. The possibly configured inversion of the input signal is considered.
Outgoing message	ES_FallingEnable	Triggered when the signal changes to the inactive state. The possibly configured inversion of the input signal is considered.
Status change	ES_BlockedChangedEnable	Triggered when the signal is set to the blocked state by the deflutter function or
Cyclic event	ES_CyclicDiEnable ES_CyclicDiInterval	Triggered by a timer with the configured interval. Serves for the cylical transmission of the digital signals.

The events for the object protection function for incoming/outgoing walk-in/alarm messages are automatically activated when the object protection function is activated.

Parameters: Event Settings Section

11.1.1.2 Event for counter

Counter values are transmitted exclusively cyclically. There is therefore only one configurable cyclic event for all counters together. Different intervals in the range from every minute to every month can be configured. The cyclic timer event is synchronised with the internal time. This enables regular transmission of the counter values at a specific time/date.

Time synchronisation makes it possible to use the transmitted metered values, even from several Clueys, for balancing. E.g. in energy/consumption monitoring applications.

11.1.1.2.1 Parameters for counter transmission interval

Function	Parameters for enable/disable	Description
Enable/disable	ES_CyclicCntEnable	Switching the cyclic counter value transmission on or off
Interval monthly or weekly	ES_CyclicCntTimeDateWeekDaySel	Changing between weekly and monthly transmission interval
Day of transmission (DD)	ES_CyclicCntTimeDateWeekDay	Day of transmission for daily (=0), weekly (=weekday: 1..7) or monthly transmission (=1...31)
Hour for transmission time (HH)	ES_CyclicCntTimeHour	Sets the time for the transmission
Minute for transmission time (MM)	ES_CyclicCntTimeMinute	
Minute interval	CS_CyclicCntTimeInterval	Interval in minutes for multiple transmissions per day

11.1.1.2.2 Monthly counter value transmission

For a counter value transmission once a month, the parameters must be set as follows:

Parameter setting	Description
ES_CyclicCntTimeDateWeekDaySel=0	Day of month (not day of week) selected
ES_CyclicCntTimeDateWeekDay="DD"	DD= Day of the month(1...31) on which to transmit
ES_CyclicCntTimeHour="HH"	HH= Hour of transmission time
ES_CyclicCntTimeMinute="MM"	MM = minute of the transmission time
ES_CyclicCntTimeInterval=0	No minute interval

The count value transmission thus takes place every month on DD. at HH:MM.

11.1.1.3 Weekly count value transmission

For a counter value transmission once a week, the parameters are to be set as follows:

Parameter setting	Description
ES_CyclicCntTimeDateWeekDaySel=1	Weekday selected
ES_CyclicCntTimeDateWeekDay="WD"	WD= Weekday(1...7) on which to transmit
ES_CyclicCntTimeHour="HH"	HH= Hour of transmission time
ES_CyclicCntTimeMinute="MM"	MM = minute of the transmission time
ES_CyclicCntTimeInterval=0	No minute interval

The count value transmission thus takes place on the WD of the week at HH:MM.

11.1.1.4 Once daily count value transmission

For a counter value transmission once a day, the parameters are to be set as follows:

Parameter setting	Description
ES_CyclicCntTimeDateWeekDaySel=0	Setting not relevant
ES_CyclicCntTimeDateWeekDay=0	0→ daily transmission
ES_CyclicCntTimeHour="HH"	HH= Hour of transmission time
ES_CyclicCntTimeMinute="MM"	MM = minute of the transmission time
ES_CyclicCntTimeInterval=0	No minute interval

The count value is thus transmitted every day at HH:MM.

11.1.1.4.1 Multiple daily count value transmission

For a multiple daily count value transmission, the parameters are to be set as follows:

Parameter setting	Description
ES_CyclicCntTimeDateWeekDaySel=0	Setting not relevant
ES_CyclicCntTimeDateWeekDay=0	0→ daily transmission
ES_CyclicCntTimeHour="HH"	HH= Hour of transmission time
ES_CyclicCntTimeMinute="MM"	MM = minute of the transmission time
ES_CyclicCntTimeInterval="MI"	Interval in minutes

The count value transmission takes place at intervals of "MI" minutes, synchronised at the time HH:MM. This means that at "HH": "MM" and all "MI" minutes before and after this time.

Example:

Time: HH:MM = 15:05 ; Interval MI=25 minutes

The transmission then takes place at the following times:

00:05	First transmission of the day (synchronisation time - N* interval)	
00:30		
....		
14:50	Synchronisation time	- interval
15:05	Synchronisation time	
16:20	Synchronisation time	+ interval
16:45	Synchronisation time	+ 2 *Interval
....		
....		
23:25		
23:50	Last transmission of the day (synchronisation time + N* interval)	

00.05	First transmission of the following day	
.....		

The synchronisation to the configured time refers to the current day in each case. If the interval is not a whole fraction of 24 hours, the time interval between the last transmission of the day and the first transmission of the following day is shorter than the configured interval.

11.1.1.5 Events for analogue value transmission

The following events of analogue values can trigger a transmission if the corresponding

Event	Parameters for enable/disable	Description
Upcoming Limit1 Violation	AES_Limit1RisingEnable	Triggered by upcoming violation of limit value 1 i.e. when setting the LIMIT1 bit
Upcoming Limit2 Violation	AES_Limit2RisingEnable	triggered in case of upcoming violation of limit value 2 i.e. when setting the LIMIT2 bit
On-going limit value1 violation	AES_Limit1FallingEnable	Triggered in the event of an outright violation of limit value 1 i.e. when resetting the LIMIT1 bit
On-going limit value2 violation	AES_Limit2FallingEnable	Triggered in the event of a partial violation of the limit value 2 i.e. when resetting the LIMIT2 bit
Delta Event	AES_DeltaEnable	triggered when the analogue value has changed by a certain value compared to the last transmission
Wire break/ Underrange	AES_InvalidValueEnable	triggered by a change of the limit value state, i.e. the "INVALID" bit

Overrange/ Sensor error detection	AES_OverflowValueEnable	triggered when the limit value state, i.e. the "OVERFLOW" bit, changes.
Cyclic event	CS_CyclicAiEnable CS_CyclicAiInterval	Triggered by a timer with the configured interval. Serves for the cylical transmission of the digital signals.

11.1.2 Data transmission

The transmission of the captured data is triggered by events, as described above, cyclically or triggered by changes.

The data objects of the digital, the counter data and analogue value objects to be transmitted are each written into a transmit buffer when an event occurs and from there packed into one or more data packets and transmitted.

The structure of the data packet is dynamic. This means that only those data objects are entered that are enabled and for which an event has occurred.

Alternatively, a static data packet format can be configured. This has a fixed structure that is always the same. It only contains the enabled inputs, counter values and analogue values.


Parameter: "PayloadFormat

11.1.3 Transmission delay

In order to make the data transmission efficient and thus not to transmit individual data packets with only one object each time a digital event occurs, a transmission delay can be configured.


Parameter: "ES_Delay

After the occurrence of a first event (empty send buffer), the delay time is waited for. Any subsequent events that occur during this time are collected in the send buffer. Sending is started after the delay time has elapsed, so that several data objects generated by successive events can be sent together in one package.

However, in order to send important events, such as limit violations, immediately without delay, the corresponding digital input can be prioritised by configuration.


Parameter: "ES_Priority

11.1.4 Transmission Confirmed/Unconfirmed

For the data packet of the digital input events and the meter events, it can be individually selected whether the transmission is confirmed or unconfirmed.


Parameters: "ES_DiConfirmed", "ES_CntConfirmed

In the case of "confirmed" transmission, the data packets are sent repeatedly if the network server does not confirm receipt. The maximum number of repetitions can be configured.


Parameter: "ConfirmedTries

To improve the change for a successful transmission, the data rate is reduced every second transmission repetition.

For other data packets, except those with digital or counter or analogue value events, the setting of the parameter: "ConfirmedTx" applies with regard to confirmed/unconfirmed transmission.

Transmissions triggered by pressing a key, transmissions after a reset or data packets triggered by a power interruption are always transmitted confirmed.

11.2 Time synchronisation

The corresponding function of the LoRaWAN® protocol is used to synchronise the time of the Clueys real-time clock.

For this purpose, a TimeSync request is sent in the next LoRa telegram to be sent, which is answered by the LoRaWAN® network server with the current time information, whereby the Cluey sets the real-time clock accordingly.

After restarting the Cluey, the time is requested in the first data telegram. This is usually an info telegram.

In addition, the Cluey can cyclically request the current time information. The parameter

TimeSyncInterval

with which the interval can be set in units of hours.

12 Configuration

The Cluey is configured via the configuration file, which can be accessed via the USB interface. In addition, the configuration parameters can be read out and also changed via the LoRaWAN® connection.

When writing the file, the parameters are taken over. If necessary, a restart of the Cluey is executed.

12.1 Configuration file "cfg.txt" and parameter description

Structure of the configuration file:

Line	Parameter	Comment	Description
1	App.vers.:3.02		Read-only: Firmware version
2	Cluey AM		Read-only: Device type
3			
4	LoRaWAN® 1.0.3rA EU868 Params (LoRaMac version 444):		Read-only: LoRaWAN® -Version and Regional Parameters
5			
6	Activation:		
7	OTAA=1	(0: ABP, 1: OTAA)	Registration procedure on the LoRaWAN® network
8			
9	OTAA (OverTheAirActivation):		The following are the parameters for OTAA
10	DevEUI=1650414D52004D00	(READ ONLY)	Read-only: Unique device identification number
11			
12	ABP (ActivationByPersonalization):		The following are the parameters for ABP registration
13	DevAddr=00F3AFBA		Device address according to LoRaWAN® specification, must be unique in the LoRa network).
14			
15	Datarate (0..5; SF12..SF7):		
16	ADR=1	(when ADR is off then datarate is fixed to DefDatarate)	Switch adaptive data rate on or off
17	DefDatarate=0	(used when ADR is off and in OTAA for joining)	Default data rate
18	Rx2DefDatarate=0	(only for ABP)	Data rate for the 2 receive window, if ABP is used, must match the setting in the network server.

19			
20	Communication:		
21	ConfirmedTx=1	(0:unconfirmed 1:confirmed uplinks; confirm is always on, on first uplink)	Currently not used!
22	ConfirmedTries=4	(1..8; try 3+5+7 will decrement datarate)	Maximum number of telegrams transmitted for each telegram sent if the confirmation of the network server is not received.
23	RejoinTrigger=00000	(Re-join after 0..16383 uplinks, 0 for OFF)	After the set number of uplinks, a re-join is carried out.
24	RejoinInterval=00001	(0..65535 d)	After the set number of days, a re-join is carried out.
25			
26	Device Information:		
27	ComtacDeviceType=017		Read only: Device type identifier Transmitted in the telegram header
28	ComtacDeviceVersion=003		Read only: Device version Transmitted in the telegram header
29	Label="Cluey AM "	(max. 30 chars)	Device identification, e.g. to indicate the installation location.
30			
31	Device Settings:		
32	DefaultSupplyMode=0	(0: external DC supply, 1: battery)	Operating mode setting(0): With external power supply, the unit switches to buffered mode when the power supply is disconnected. This means that the unit then runs from the battery for the time set below. In battery mode(1), the Cluey runs permanently from the internal battery and corresponding time settings.
33	BufferedOperation=0	(0: not enabled, 1: enabled)	Activate buffered mode in case of loss of external supply voltage
34	BufferedOperationSpan=00010	(0..65535 s)	Time limit of the buffered mode in case of loss of the external supply voltage
35	PayloadFormat=1	(0: static payload, 1: dynamic payload)	Selection of the payload format
36			
37	Timing Settings:		
38	MeasIntervalDcSupply=00100	(30..32000 ms)	Internal processing interval for operation with external supply voltage and buffered mode, i.e. applied when DefaultSupplyMode=0
39	MeasIntervalBattery=01000	(30..32000 ms)	Internal processing interval for battery operation i.e. applied when DefaultSupplyMode=1
40	TimeSyncInterval=00024	(0..65535 h)	Interval for requesting the time from the network server for synchronisation of the internal real-time clock
41	MeasIntervalAi=00005	(1..65535 s)	Acquisition cycle for analogue values
42			
43	Input Settings (IS):		
44	IS_Enable=1;1;1;1;1;1;1;1	(0: not enabled, 1: enabled)	Activation of the digital input function for the connections IO1, IO2,IO3,IO4,AIN5,AIN6,AIN7,AIN8
45	IS_Invert=0;0;0;0;0;0;0;0	(0: not inverted 1: inverted)	Activation of the inversion of the input states for the digital and counter inputs IO1,IO2,IO3,IO4,AIN5,AIN6,AIN7,AIN8
46	IS_Active=0;0;0;0;0;0;0;0	(0: not enabled, 1: enabled)	Selecting the function of the digital inputs IO1,IO2,IO3,IO4,AIN5,AIN6,AIN7,AIN8
47	IS_DelayEnable=1;1;1;1;1;1;1;1	(0: not enabled, 1: enabled)	Activation of the delay function for the digital inputs IO1,IO2,IO3,IO4,AIN5,AIN6,AIN7,AIN8
48	IS_DelayRising=00100;00100;00100;00100;00100;00100;00100;00100	(0..65535)	Delay time for coming message for the digital inputs IO1,IO2,IO3,IO4,AIN5,AIN6,AIN7,AIN8
49	IS_DelayFalling=00100;00100;00100;00100;00100;00100;00100;00100	(0..65535)	Delay time for outgoing message for digital inputs IO1,IO2,IO3,IO4,AIN5,AIN6,AIN7,AIN8
50	IS_DelayScaling=0;0;0;0;0;0;0;0	(0: ms, 1: s, 2: min)	Delay time scaling for the previous delay values IO1,IO2,IO3,IO4,AIN5,AIN6,AIN7,AIN8
51	IS_WiperEnable=0;0;0;0;0;0;0;0	(0: not enabled, 1: enabled)	Activation of the wiper function for the digital inputs IO1,IO2,IO3,IO4,AIN5,AIN6,AIN7,AIN8
52	IS_WiperConfirmationTimeout=00010;00010;00010;00010;00010;00010;00010;00010	(0..65535 s)	Confirmation timeout for wiper messages IO1,IO2,IO3,IO4,AIN5,AIN6,AIN7,AIN8
53	IS_DeflutterEnable=0;0;0;0;0;0;0;0	(0: not enabled, 1: enabled)	Activation of the deflutter function for the digital inputs IO1,IO2,IO3,IO4,AIN5,AIN6,AIN7,AIN8
54	IS_DeflutterInterval=00010;00010;00010;00010;00010;00010;00010;00010	(0..65535 s)	Monitoring time span for the deflutter function for the digital inputs IO1,IO2,IO3,IO4,AIN5,AIN6,AIN7,AIN8
55	IS_DeflutterCount=00005;00005;00005;00005;00005;00005;00005;00005	(0..65535)	Number of permissible edge changes within the monitoring interval for the defluttering function for the digital inputs IO1,IO2,IO3,IO4,AIN5,AIN6,AIN7,AIN8
56	IS_DoubleEnable=0;0;0;0	(0: not enabled, 1: enabled)	Activation of the double message function for the digital input pairs IO1+IO2,IO3+IO4,AIN5+AIN6,AIN7+AIN8
57	IS_DoubleIntermediateStateTimoutEnable=0;0;0;0	(0: not enabled, 1: enabled)	Activation of the intermediate position monitoring for the double message function for the digital input pairs IO1+IO2, IO3+IO4, AIN5+AIN6,AIN7+AIN8
58	IS_DoubleIntermediateStateTimout=00060;00060;00060;00060	(0..65535 s)	Intermediate timeout for the double message function for the digital input pairs IO1+IO2, IO3+IO4, AIN5+AIN6, AIN7+AIN8
59	IS_CounterEnable=0;0;0;0;0;0;0;0	(0: not enabled, 1: enabled)	Activation of the counting function for the connections IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8

60	IS_CounterMode=0;0;0;0;0;0;0;0	(0: pulse mode, 1: operating time mode)	Selection of the counter mode for the connections IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
61	IS_CounterScaling=1;1;1;1;1;1;1;1	(0: ms, 1: s, 2: min, 3: h)	Scaling for the operating time counters
62			
63	Analogue Input Settings (AIS):		
64	AIS_Enable=0;0;0;0;0;0;0;0	(0: not enabled, 1: 0-10V, 2: 0-20mA)	Activation and selection of the analogue input function for the connections IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
65	AIS_DeltaEnable=0;0;0;0;0;0;0;0	(0: not enabled, 1: enabled)	Activation of the delta event function for the analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
66	AIS_DeltaValue=00500;00500;00500;00500;00500;00500;00500;00500	(0..65535)	Delta event limit for analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
67	AIS_Limit1Enable=0;0;0;0;0;0;0;0	(0: not enabled, 1: enabled)	Activation of the limit value function 1 for the analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
68	AIS_Limit2Enable=0;0;0;0;0;0;0;0	(0: not enabled, 1: enabled)	Activation of the limit value function 2 for the analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
69	AIS_Limit1DelayEnable=0;0;0;0;0;0;0;0	(0: not enabled, 1: enabled)	Activation of the delay for limit value 1 for the analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
70	AIS_Limit2DelayEnable=0;0;0;0;0;0;0;0	(0: not enabled, 1: enabled)	Activation of the delay for limit value 2 for the analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
71	AIS_LimitDelayScaling=0;0;0;0;0;0;0	(0: ms, 1: s, 2: min, 3: h)	Scaling for the following limit value delay values (line 72 to 75)
72	AIS_Limit1DelayRising=00100;00100;00100;00100;00100;00100;00100;00100	(0..65535)	Delay time for coming limit value 1 violation for the analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8. Time unit according to line 71
73	AIS_Limit2DelayRising=00100;00100;00100;00100;00100;00100;00100;00100	(0..65535)	Delay time for coming limit value 2 violation for the analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8. Time unit according to line 71
74	AIS_Limit1DelayFalling=00100;00100;00100;00100;00100;00100;00100;00100	(0..65535)	Delay time for outgoing limit value 1 violation for analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8. Time unit according to line 71
75	AIS_Limit2DelayFalling=00100;00100;00100;00100;00100;00100;00100;00100	(0..65535)	Delay time for outgoing limit value 1 violation for analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8. Time unit according to line 71
76	AIS_Limit1Value=08000;08000;08000;08000;08000;08000;08000;08000	(0..65535)	Limit value 1 for the analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
77	AIS_Limit2Value=08000;08000;08000;08000;08000;08000;08000;08000	(0..65535)	Limit value 2 for the analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
78	AIS_Limit1Hysteresis=00500;00500;00500;00500;00500;00500;00500;00500	(0..65535)	Hysteresis for limit value 1 for analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
79	AIS_Limit2Hysteresis=00500;00500;00500;00500;00500;00500;00500;00500	(0..65535)	Hysteresis for limit value 2 for analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
80	AIS_Limit1Direction=0;0;0;0;0;0;0;0	(0: lower, 1: upper)	Selection of the limit value direction for limit value 1: Over(1)- or underrun(0)
81	AIS_Limit2Direction=0;0;0;0;0;0;0;0	(0: lower, 1: upper)	Selection of the limit value direction for limit value 2: Over(1)- or underrun(0)
82			
83	Digital Event Settings (IES):		
84	IES_RisingEnable=1;1;1;1;1;1;1;1	(0: not enabled, 1: enabled)	Activate the transmission events for the digital inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8 when a digital message is received.
85	IES_FallingEnable=1;1;1;1;1;1;1;1	(0: not enabled, 1: enabled)	Activate the transmission events on outgoing digital message for the digital inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
86	IES_BlockedChangedEnable=1;1;1;1;1;1;1;1	(0: not enabled, 1: enabled)	Activating the transmission events when the blocking state (deflutter function) of the digital message changes for the digital inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
87	IES_Priority=0;0;0;0;0;0;0;0	(0: low priority, 1: high priority)	Activate the transmission priority, for the digital events. When activated for the corresponding input, the following delay time is not considered
88	IES_Delay=00005	(0..65535 s)	Transmission delay after occurrence of a digital event
89			
90	Analogue Event Settings (AES):		
91	AES_Limit1RisingEnable=1;1;1;1;1;1;1;1	(0: not enabled, 1: enabled)	Activate the transmission events for the inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8 in the event of a limit value 1 violation.
92	AES_Limit2RisingEnable=1;1;1;1;1;1;1;1	(0: not enabled, 1: enabled)	Activate the transmission events for the inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8 in the event of a limit value 2 violation.
93	AES_Limit1FallingEnable=1;1;1;1;1;1;1;1	(0: not enabled, 1: enabled)	Activate the transmission events on going limit value 1 violation for inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
94	AES_Limit2FallingEnable=1;1;1;1;1;1;1;1	(0: not enabled, 1: enabled)	Activate the transmission events on going limit value 2 violation for inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
95	AES_DeltaEnable=1;1;1;1;1;1;1;1	(0: not enabled, 1: enabled)	Activate transmission events on delta event for inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
96	AES_InvalidValueEnable=1;1;1;1;1;1;1;1	(0: not enabled, 1: enabled)	Activate the transmission events on invalid event for inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
97	AES_OverflowValueEnable=1;1;1;1;1;1;1;1	(0: not enabled, 1: enabled)	Activate transmission events on overflow event for inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
98			
99	Cyclic Settings (CS):		

100	CS_CyclicDiEnable=1;1;1;1;1;1;1;1	(0: not enabled, 1: enabled)	Activate cyclic transmission of the digital input states IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
101	CS_DiConfirmed=0	(0: not enabled, 1: enabled)	Enable "confirmed" transmission for digital events IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
102	CS_CyclicDiInterval=03600;03600;03600;03600;03600;03600;03600;03600	(1..65535 s)	Interval for cyclical transmission Digital input states IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
103	CS_CyclicAiEnable=1;1;1;1;1;1;1;1	(0: not enabled, 1: enabled)	Activate cyclic transmission of the analogue input values IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
104	CS_AiConfirmed=0	(0: not enabled, 1: enabled)	Enable "confirmed" transmission for analogue events IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
105	CS_CyclicAiInterval=03600;03600;03600;03600;03600;03600;03600;03600	(1..65535 s)	Interval for cyclical transmission Analogue input values IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
106	CS_CyclicCntEnable=1	(0: not enabled, 1: enabled)	Activate the cyclical transmission of the counter values IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
107	CS_CntConfirmed=0	(0: not enabled, 1: enabled)	Activate "confirmed" transmission for counter values IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
108	CS_CyclicCntTimeDateWeekDaySel=0	(0: Date, 1: WeekDay)	Selection: Weekly or daily interval
109	CS_CyclicCntTimeDateWeekDay=00	(0..7, 0: every day, 1: monday,.. OR 0..31, 0: every day, 1: 1st of month,..)	Selecting the day of the week or month for cyclical transmission
110	CS_CyclicCntTimeHour=04	(0..23)	Transmission time : Hour
111	CS_CyclicCntTimeMinute=05	(0..23)	Transmission time : Minute
112	CS_CyclicCntTimeInterval=00480	(0: day of week/month, 1..1440 min: daily)	Transmission cycle for daily transmission
113			
114	Access Control Settings (ACS):		
115	ACS_Enable=0	(0: not enabled, 1: enabled)	Activation of the object protection function
116	ACS_AlarmDelay=00120	(0..65535 s)	Delay time for the object protection function
117	ACS_MotionDetectorSel=1	(0: not enabled, 1..8: channel number)	Selecting the input for the motion detector for the object protection function
1198	ACS_KeySwitchSel=2	(0: not enabled, 1..8: channel number)	Selection of the input for the key switch for the object protection function
119	ACS_DoorContactSel=3	(0: not enabled, 1..8: channel number)	Selection of the input for the door contact for the object protection function
120			
121	Output Settings (OS):		
122	OS_Enable=0;0;0;0	(0: not enabled, 1: enabled)	Activation of the digital output function for the connections IO1, IO2, IO3, IO4
123	OS_Invert=0;0;0;0	(0: not inverted 1: inverted)	Activation of the inversion of the digital output function for the connections IO1, IO2, IO3, IO4
124	OS_Mode=0;0;0;0	(0: static, 1: wiper)	Selection of the function for the outputs IO1, IO2, IO3, IO4
125	OS_WiperTime=00150;00150;00150;00150	(0..65535 ms)	Wipe duration for the wiper function of outputs IO1, IO2, IO3, IO4
126	OS_TriggerEnable=1	(0: not enabled, 1: enabled)	Activating the trigger function
127	OS_TriggerPeriod=30	(1..65535 s)	Trigger interval
128	OS_TriggerPeriodScaling=0	(0: s, 1: min, 2: h)	Scaling for the trigger interval
127			
128	IO Mapping Settings (IOMS):		
129	_____|DI_____|ACS_____		
130	IOMS_Out1=0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0	(Output control by DI: 0: not enabled, 1: set, 2: reset , 3: follow)	Selection of the control mode for output IO1 by digital inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8, as well as internal messages (object protection, trigger)
131	IOMS_Out2=0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0	(Output control by DI: 0: not enabled, 1: set, 2: reset , 3: follow)	Selection of the control mode for output IO2 by digital inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8, as well as internal messages (object protection, trigger)
132	IOMS_Out3=0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0	(Output control by DI: 0: not enabled, 1: set, 2: reset , 3: follow)	Selection of the control mode for output IO3 by digital inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8, as well as internal messages (object protection, trigger)
133	IOMS_Out4=0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0	(Output control by DI: 0: not enabled, 1: set, 2: reset , 3: follow)	Selection of the control mode for output IO4 by digital inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8, as well as internal messages (object protection, trigger)
134			
135	Limit-Output Mapping Settings (LOMS):		
136	_____|LIMIT1_____|LIMIT2_____		
137	LOMS_Out1=0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0	(Output control by Limit: 0: not enabled, 1: set, 2: reset , 3: follow)	Selection of the control mode for output IO1 by limit values 1 and 2 of the analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
138	LOMS_Out2=0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0	(Output control by Limit: 0: not enabled, 1: set, 2: reset , 3: follow)	Selection of the control mode for output IO2 by limit values 1 and 2 of the analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8
139	LOMS_Out3=0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0	(Output control by Limit: 0: not enabled, 1: set, 2: reset , 3: follow)	Selection of the control mode for output IO3 by limit values 1 and 2 of the analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8

140	LOMS_Out4=0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0	(Output control by Limit: 0: not enabled, 1: set, 2: reset , 3: follow)	Selection of the control mode for output IO4 by limit values 1 and 2 of the analogue inputs IO1, IO2, IO3, IO4, AIN5, AIN6, AIN7, AIN8

The values shown here are the value settings

In the listing, the values from left to right refer to the indices from 1 to...., e.g. input 1 to 8

13 Message formats

The following shows the structure of the messages transmitted via the LoRa radio interface in send (uplink) and receive (downlink) direction.

13.1 General information on the data formats

13.1.1 ObjectType and ObjectId

For the identification of the data, the various data objects are described with an ObjectType and ObjectId as follows.

ObjectType designates the type of object, e.g. digital values. The ObjectId designates the number of the associated input or output, or refers to an internally created object (e.g. object protection messages).

13.1.1.1 Object type: Digital values (singlePointInfo)

The ObjectType =01 is used for digital values (singlePointInfo). The ObjectId is assigned as follows:

ObjectType	ObjectId	Description
0x01	0	16 digital values packed into one data word The bit position corresponds to the ObjectId of the individual digital values
	1	Digital input 1 (IO1)
	2	Digital input 2 (IO2)
	3	Digital input 3 (IO3)
	4	Digital input 4 (IO4)
	5	Digital input 5 (IN5)
	6	Digital input 6 (IN6)
	7	Digital input 7 (IN7)
	8	Digital input 8 (IN8)
	9	Walk-in report of the object protection function
	10	Alarm message of the object protection function
	11	reserved
	12	reserved
	13	reserved
	14	reserved

	15	reserved
	16	reserved

13.1.1.2 Object type: Double messages (doublePointInfo)

The ObjectType =02 is used for double messages (doublePointInfo). The ObjectId is assigned as follows:

ObjectType	ObjectId	Description
0x02	1	Double message 1 (IO1 and IO2)
	2	Double message 2 (IO3and IO4)
	3	Double message 3 (IN5and IN6)
	4	Double message 4 (IN7and IN8)

13.1.1.3 Object type: Counter values

The ObjectType =04 is used for counter values. The ObjectId is assigned as follows:

ObjectType	ObjectId	Description
0x04	1	Counter value 1 (IO1)
	2	Count value 2 (IO2)
	3	Counter value 3 (IO3)
	4	Counter value 4 (IO4)
	5	Counter value 5 (IN5)
	6	Counter value 6 (IN6)
	7	Counter value 7 (IN7)
	8	Counter value 8 (IN8)

13.1.1.4 Object type: Analogue values

The ObjectType =05 is used for counter values. The ObjectId is assigned as follows:

ObjectType	ObjectId	Description
0x05	1	Analogue value 1 (IO1)
	2	Analogue value 2 (IO2)
	3	Analogue value 3 (IO3)
	4	Analogue value 4 (IO4)
	5	Analogue value 5 (IN5)
	6	Analogue value 6 (IN6)
	7	Analogue value 7 (IN7)
	8	Analogue value 8 (IN8)

13.2 Uplink Messages

13.2.1 Uplink Message Header

Upload messages begin with a header that can be used to identify the device and recognise the state of the Cluey.

<table>
<tr><th>Content</th><th>Structure</th><th>Length</th><th>Description</th></tr>
<tr><td>Comtac Device ID</td><td>
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>4</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>E</td><td colspan="7">DeviceId</td></tr>
</table>
Cluey DeviceID = 17
</td><td>1 byte</td><td>Unique type ID for Comtac units
E is reserved for later extensions and is here always =0</td></tr>
<tr><td>Comtac Device Version</td><td>
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>4</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="7">DeviceVersion</td></tr>
</table>
</td><td>1 byte</td><td>Version of the Comtac Device.
Serves together with the device ID to assign the appropriate payload decoder.
(0...255)</td></tr>
<tr><td>Device Status</td><td>
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>UC</td><td>LV</td><td>RST</td><td>TC</td><td>BP</td><td>TS</td><td>BO</td><td>CE</td></tr>
</table>
CE: Configuration Error (Error in Configuration downlink-Command)
BO: Transmit Buffer overflow
TS: clock sync'ed
BP: Battery Powered
TC: transmission credits consumed
RST: Restart of device
LV: Low Voltage
UC: confirmation timeout (telegram before actual one)
</td><td>1 byte</td><td></td></tr>
<tr><td>Battery Voltage</td><td>
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="7">Battery capacity</td></tr>
</table>
0...100 (%)
255: Error (>3.3V)
0: DC powered
</td><td>1 byte</td><td></td></tr>
</table>

13.2.2 Static data packet (uplink port 3)

The static data packet is intended for simple decoding and is sent instead of the dynamic data packets if this is specified with the parameter

"PayloadFormat",

was selected accordingly.

In the data packet, the position of the information is always the same.

The disadvantage of this is that all information is always transmitted, even if it has not changed, so that transmission time may be unnecessarily consumed.

The telegram has a fixed structure that depends only on the configuration. This telegram is sent when an event occurs, when a key is pressed or a request is made via downlink). Regardless of which event has triggered the transmission, all data - digital, counter and analogue values - are always sent.

Structure of the static data packet:

Source/Content	Structure	Size [bytes]	Remark
Message Header Device info	Uplink Message Header	4	

<table>
<tr><td>Timestamp</td><td>
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="8">UNIX Timestamp MSB</td></tr>
<tr><td>2</td><td colspan="8">UNIX Timestamp</td></tr>
<tr><td>3</td><td colspan="8">UNIX Timestamp</td></tr>
<tr><td>4</td><td colspan="8">UNIX Timestamp LSB</td></tr>
</table>
</td><td>4</td><td></td></tr>
<tr><td>Digital values:
Digital input 1...8 and internally formed digital values 9...16</td><td>
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td rowspan="2">1</td><td colspan="4">COT</td><td colspan="4">Status (not used)</td></tr>
<tr><td>CYC</td><td>INT</td><td>EVT</td><td>0</td><td>0</td><td>BL</td><td>D1</td><td>D0</td></tr>
<tr><td>2</td><td colspan="8">DI 16...9</td></tr>
<tr><td>3</td><td colspan="8">DI 8...1</td></tr>
</table>
DI8...DI1: State of the digital inputs DI8...DI1

DI16...9: State of the internally formed digital values

- DI9: Walk-in message of the object protection function

- DI10: Alarm message of the object protection function

- DI11...16: reserved for future applications

COT: Cause of Transmission:
- CYC: Cyclic event
- INT: Interrogation (Button, Downlink)
- EVT: Event (change of state)
</td><td>3</td><td></td></tr>
<tr><td>Input 1: Analogue value 1 or counter value 1</td><td>
For analogue value:
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>4</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td rowspan="2">1</td><td colspan="4">ObjectType</td><td colspan="4">ObjectNo</td></tr>
<tr><td colspan="8">0x51</td></tr>
<tr><td rowspan="2">2</td><td colspan="4">COT</td><td colspan="4">Status</td></tr>
<tr><td>CYC</td><td>INT</td><td>EVT</td><td>LIM</td><td>IV</td><td>OV</td><td>LIM2</td><td>LIM2</td></tr>
<tr><td>3</td><td colspan="8">0</td></tr>
<tr><td>4</td><td colspan="8">Analogue value MSB</td></tr>
<tr><td>5</td><td colspan="8">Analogue value LSB</td></tr>
</table>
COT: Cause of transmission
- CYC: Cyclic event
- INT: Interrogation triggered by button, downlink or first transmission after restart or rejoin
- EVT: Event, e.g. Delta Event
- LIM: Limit value over/underrun
Status:
- IV: invalid, below measuring range
- OV: overflow, measuring range exceeded
- LIM1: Limit value1 violated
- LIM2: Limit value2 violated

For count values:
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>4</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td rowspan="2">1</td><td colspan="4">ObjectType</td><td colspan="4">ObjectNo</td></tr>
<tr><td colspan="8">0x41</td></tr>
<tr><td rowspan="2">2</td><td colspan="4">COT</td><td colspan="4">Status</td></tr>
<tr><td>CYC</td><td>INT</td><td>EVT</td><td>LIM</td><td>0</td><td>0</td><td>RES</td><td>OV</td></tr>
<tr><td>3</td><td colspan="8">Count MSB</td></tr>
<tr><td>4</td><td colspan="8">Count</td></tr>
<tr><td>5</td><td colspan="8">Count LSB</td></tr>
</table>
COT: Cause of transmission
- CYC: Cyclic event
- INT: Interrogation triggered by button, downlink, or at first transmission after restart or rejoin
- EVT: not used
- LIM: not used
Status:
- OV: counter overflow
- RES: Counter reset (since last transmission)
</td><td>5</td><td></td></tr>
</table>

Input 2: Analogue value 2 or counter value 2	As above, but ObjectNo =2	5	
Input 3: Analogue value 3 or counter value 3	As above, but ObjectNo =3	5	
Input 4: Analogue value 4 or counter value 4	As above, but ObjectNo =4	5	
Input 5: Analogue value 5 or counter value 5	As above, but ObjectNo =5	5	
Input 6: Analogue value 6 or counter value 6	As above, but ObjectNo =6	5	
Input 7: Analogue value 7 or counter value 7	As above, but ObjectNo =7	5	
Input 8: Analogue value 8 or counter value 8	As above, but ObjectNo =8	5	

The size of the data packet is 4 (header)+4 (time stamp) +3 (digital values)+8*5 (analogue/counter values) =51 bytes.

13.2.3 **Dynamic digital message data packet (uplink port 20)**

This data packet is sent if the payload format "dynamic Payload" was selected in the configuration.

It contains the states of the digital values, i.e. the digital inputs and the internally formed digital values, such as the alarm and walk-in message of the object protection function.

The data packet is sent when
- an event, i.e. a change of state,
- cyclically at the configured interval
- After restarting the Cluey or after re-join
- on request by a downlink command (interrogation)

The corresponding events must be activated in the configuration. The data packet only contains the digital values that are active and only those for which an event has occurred.

The data packet contains the device header at the beginning, followed by a reference timestamp and the data objects belonging to the events that occurred. The timestamps of the individual data objects are coded as an offset to the reference timestamp.
Each digital value data object contains, in addition to the value and time stamp, the transmission cause and the status information.

Parameter: "PayloadFormat

<table>
<tr><th>Content</th><th>Structure</th><th>Size [bytes]</th><th>Remark</th></tr>
<tr><td>Message Header</td><td>Uplink Message Header</td><td>4</td><td></td></tr>
<tr><td>Absolute Timestamp</td><td>
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="8">UNIX Timestamp MSB</td></tr>
<tr><td>2</td><td colspan="8">UNIX Timestamp</td></tr>
<tr><td>3</td><td colspan="8">UNIX Timestamp</td></tr>
<tr><td>4</td><td colspan="8">UNIX Timestamp LSB</td></tr>
</table>
</td><td>4</td><td>Absolute timestamp of the oldest event as base for recalculate if the relative timestamps of the following event objects</td></tr>
</table>

<table>
<tr>
<td>Digital Event Object (oldest in queue)</td>
<td>
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>4</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="4">ObjectType</td><td colspan="4">ObjectNo</td></tr>
<tr><td rowspan="2">2</td><td colspan="4">COT</td><td colspan="4">Status</td></tr>
<tr><td>CYC</td><td>INT</td><td>EVT</td><td>0</td><td></td><td>BL</td><td>D1</td><td>D0</td></tr>
<tr><td>3</td><td colspan="8">TimeStampOffset MSB</td></tr>
<tr><td>4</td><td colspan="8">TimeStampOffset LSB</td></tr>
</table>
ObjectType:
- 1: singlePointInfo
- 2: doublePointInfo
- 4: accessControll
ObjectNo:
- No of Input 0...7
COT: Cause of Transmission:
- CYC: Cyclic event
- INT: Interrogation (Button, downlink)
- EVT: Event (change of state)
Status:
- BL: Blocked
- D0: State of Input
- D1: for DoublePointInfo : State of associated second input, otherwise 0
</td>
<td>4</td>
<td>TimeStampOffset will be 0, because it's similar to the suceeding absolute time</td>
</tr>
<tr>
<td>More Digital Event Objects (newer)</td>
<td>See above</td>
<td>4</td>
<td>More Events in message if more in the event queue
TimestampOffset is related to Timestamp</td>
</tr>
</table>

The digital values must be included in the data packet in chronological order, not sorted by object number. A digital value (equal to object number) can also be contained several times in a data packet, e.g. if the input state changes faster than can be transmitted.

Example telegram:

Base64: EQMEAWPQL0QRIQAAEiEAABMhAAARIAAFEiAABRMgAAU=

Hex: 1103040163d02f4411210000122100001321000011200005122000051320005

Decoded Payload(JSON):


```
{
  "data" : {
   "decoder" : {
    "info" : "comtac Cluey",
    "version" : "00.07"
   },
   "payload" : {
    "data" : {
     "digitalInputs" : [ {
      "cot" : {
       "cyclic" : false,
       "event" : true,
       "interrogation" : false,
       "limit" : false
      },
      "info" : {
       "id" : 1,
       "type : singlePointInfo
      },
      "status" : {
       "blocked" : false,
       "state" : 1
      },
      "timestamp" : {
       "string" : "Tue, 24 Jan 2023 19:19:32 GMT",
       "unix" : 1674587972
```

```
  }
 }, {
  "cot" : {
   "cyclic" : false,
   "event" : true,
   "interrogation" : false,
   "limit" : false
  },
  "info" : {
   "id" : 2,
   "type : singlePointInfo
  },
  "status" : {
   "blocked" : false,
   "state" : 1
  },
  "timestamp" : {
   "string" : "Tue, 24 Jan 2023 19:19:32 GMT",
   "unix" : 1674587972
  }
 }, {
  "cot" : {
   "cyclic" : false,
   "event" : true,
   "interrogation" : false,
   "limit" : false
  },
  "info" : {
   "id" : 3,
   "type : singlePointInfo
  },
  "status" : {
   "blocked" : false,
   "state" : 1
  },
  "timestamp" : {
   "string" : "Tue, 24 Jan 2023 19:19:32 GMT",
   "unix" : 1674587972
  }
 }, {
  "cot" : {
   "cyclic" : false,
   "event" : true,
   "interrogation" : false,
   "limit" : false
  },
  "info" : {
   "id" : 1,
   "type : singlePointInfo
  },
  "status" : {
   "blocked" : false,
   "state" : 0
  },
  "timestamp" : {
   "string" : "Tue, 24 Jan 2023 19:19:37 GMT",
   "unix" : 1674587977
  }
 }, {
  "cot" : {
   "cyclic" : false,
   "event" : true,
   "interrogation" : false,
   "limit" : false
  },
  "info" : {
   "id" : 2,
   "type : singlePointInfo
  },
  "status" : {
   "blocked" : false,
   "state" : 0
```

```
      },
      "timestamp" : {
        "string" : "Tue, 24 Jan 2023 19:19:37 GMT",
        "unix" : 1674587977
      }
     }, {
      "cot" : {
        "cyclic" : false,
        "event" : true,
        "interrogation" : false,
        "limit" : false
      },
      "info" : {
        "id" : 3,
        "type : singlePointInfo
      },
      "status" : {
        "blocked" : false,
        "state" : 0
      },
      "timestamp" : {
        "string" : "Tue, 24 Jan 2023 19:19:37 GMT",
        "unix" : 1674587977
      }
     } ]
    },
    "device" : {
     "batteryLevel" : 0,
     "deviceStatus" : {
      "batteryPowered" : false,
      "bufferOverflow" : false,
      "configurationError" : false,
      "confirmationTimeout" : false,
      "deviceRestarted" : false,
      "lowSupplyVoltage" : false,
      "timeSynced" : true,
      "txCreditsConsumed" : false
     },
     "info" : {
      "deviceDesignation" : "Cluey-AM",
      "deviceId" : 17,
      "deviceManufacturer" : "comtac AG",
      "deviceVersion" : 3
     }
    }
   },
   "payloadLength" : 32,
   "port" : 20,
   "portFunction" : "DI_DATA"
  },
  "warnings" : [ ]
 }
}
```


13.2.4 **Dynamic counter data packet (uplink port 21)**

This data packet is sent if the payload format "dynamic payload" was selected in the configuration.

It contains the device header at the beginning, followed by the blocks of counter values that each belong to an event. A data packet can contain several blocks (i.e. counting events). A block always contains all count values activated in the configuration. All count values are always transmitted together, so that it is ensured,
A common time stamp and the associated counter readings are transmitted for each event (block).
Each counter object contains, in addition to the value, the transmission cause and the status information

Parameter: "PayloadFormat

<table>
<tr><th>Content</th><th>Structure</th><th>Size [bytes]</th><th>Remark</th></tr>
<tr><td>Message Header</td><td>Uplink Message Header</td><td>4</td><td></td></tr>
<tr><td>Object Count</td><td>N1=1...8</td><td>1</td><td>Number of Objects Related in the following Objectset</td></tr>
<tr><td>Common AbsoluteTimestamp</td><td>
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="8">UNIX Timestamp LSB</td></tr>
<tr><td>2</td><td colspan="8">UNIX Timestamp</td></tr>
<tr><td>3</td><td colspan="8">UNIX Timestamp</td></tr>
<tr><td>4</td><td colspan="8">UNIX Timestamp MSB</td></tr>
</table>
</td><td>4</td><td>Absolute timestamp of the oldest event</td></tr>
<tr><td>Counter Object (oldest)</td><td>
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>4</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="4">ObjectType</td><td colspan="4">ObjectNo</td></tr>
<tr><td>2</td><td colspan="4">COT</td><td colspan="4">Status</td></tr>
<tr><td></td><td>CYC</td><td>INT</td><td>EVT</td><td>LIM</td><td>0</td><td>0</td><td>RES</td><td>OV</td></tr>
<tr><td>3</td><td colspan="8">Count MSB</td></tr>
<tr><td>4</td><td colspan="8">Count</td></tr>
<tr><td>5</td><td colspan="8">Count LSB</td></tr>
</table>
COT: Cause of transmission
- CYC: Cyclic event
- INT: Interrogation triggered by button, downlink command or first transmission after restart or rejoin
- EVT: not used
- LIM: not used
Status:
- OV: Meter overflow
- RES: Counter reset (since last transmission)
</td><td>5</td><td>TimeStampOffset will be 0</td></tr>
<tr><td>More Counter Objects (newer)..N</td><td>See above</td><td>(N1-1)*5</td><td></td></tr>
<tr><td>Object Count</td><td>N2=1...8</td><td>1</td><td>Number of Objects Related in the following Objectset</td></tr>
<tr><td>Common Absolute Timestamp</td><td>
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="8">UNIX Timestamp MSB</td></tr>
<tr><td>2</td><td colspan="8">UNIX Timestamp</td></tr>
<tr><td>3</td><td colspan="8">UNIX Timestamp</td></tr>
<tr><td>4</td><td colspan="8">UNIX Timestamp LSB</td></tr>
</table>
</td><td>4</td><td>Absolute timestamp of the oldest event</td></tr>
<tr><td>Counter Object (oldest)</td><td>
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>4</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="4">ObjectType</td><td colspan="4">ObjectNo</td></tr>
<tr><td>2</td><td colspan="4">COT</td><td colspan="4">Status</td></tr>
<tr><td></td><td>RES</td><td>CTP</td><td>CTS</td><td>CTI</td><td>RES</td><td>RES</td><td>OP</td><td>OV</td></tr>
<tr><td>3</td><td colspan="8">Count LSB</td></tr>
<tr><td>4</td><td colspan="8">Count</td></tr>
<tr><td>5</td><td colspan="8">Count MSB</td></tr>
</table>
</td><td>5</td><td>TimeStampOffset will be 0</td></tr>
<tr><td>More Counter Objects (newer)..N</td><td>See above</td><td>(N2-1)*5</td><td></td></tr>
</table>

Example telegram:

Base64: EQMEAQNj0DBQQYAAANRCgAAA1EOAAADU

Hex: 110304010363d0305041800000d442800000d443800000d4

Decoded Payload(JSON):


```
{
  "data" : {
   "decoder" : {
     "info" : "comtac Cluey",
     "version" : "00.07"
```

```
},
"payload" : {
 "data" : {
  "counters" : [ {
   "cot" : {
    "cyclic" : true,
    "event" : false,
    "interrogation" : false,
    "limit" : false
   },
   "info" : {
    "id" : 1,
    "type" : "counter"
   },
   "status" : {
    "limit" : false,
    "overflow" : false,
    "reset" : false
   },
   "timestamp" : {
    "string" : "Tue, 24 Jan 2023 19:24:00 GMT",
    "unix" : 1674588240
   },
   "value" : 212
  }, {
   "cot" : {
    "cyclic" : true,
    "event" : false,
    "interrogation" : false,
    "limit" : false
   },
   "info" : {
    "id" : 2,
    "type" : "counter"
   },
   "status" : {
    "limit" : false,
    "overflow" : false,
    "reset" : false
   },
   "timestamp" : {
    "string" : "Tue, 24 Jan 2023 19:24:00 GMT",
    "unix" : 1674588240
   },
   "value" : 212
  }, {
   "cot" : {
    "cyclic" : true,
    "event" : false,
    "interrogation" : false,
    "limit" : false
   },
   "info" : {
    "id" : 3,
    "type" : "counter"
   },
   "status" : {
    "limit" : false,
    "overflow" : false,
    "reset" : false
   },
   "timestamp" : {
    "string" : "Tue, 24 Jan 2023 19:24:00 GMT",
    "unix" : 1674588240
   },
   "value" : 212
  } ]
 },
 "device" : {
  "batteryLevel" : 0,
  "deviceStatus" : {
   "batteryPowered" : false,
```

```
      "bufferOverflow" : false,
      "configurationError" : false,
      "confirmationTimeout" : false,
      "deviceRestarted" : false,
      "lowSupplyVoltage" : false,
      "timeSynced" : true,
      "txCreditsConsumed" : false
     },
     "info" : {
      "deviceDesignation" : "Cluey-AM",
      "deviceId" : 17,
      "deviceManufacturer" : "comtac AG",
      "deviceVersion" : 3
     }
    }
   },
   "payloadLength" : 24,
   "port" : 21,
   "portFunction" : "CNT_DATA"
  },
  "warnings" : [ ]
}
```


13.2.5 Dynamic analogue value data packet (uplink port 23)

This data packet is sent if the payload format "dynamic Payload" was selected in the configuration.

It contains the analogue values of the inputs that are configured as analogue inputs.

The data packet is sent when

- an event has occurred, i.e. in the case of a limit value violation, delta event, overrange or underrange
- cyclically at the configured interval
- After restarting the Cluey or after re-join
- on request by a downlink command (interrogation)

The corresponding events must be activated in the configuration. The data packet only contains the analogue values of the inputs that are active and only those for which an event has occurred.

The data packet contains the device header at the beginning, followed by a reference timestamp and the data objects belonging to the events that occurred. The timestamps of the individual data objects are coded as an offset to the reference timestamp.
Each analogue value data object contains, in addition to the value and time stamp, the transmission cause and the status information.

Parameter: "PayloadFormat

<table>
<tr><th>Content</th><th>Structure</th><th>Size [bytes]</th><th>Remark</th></tr>
<tr><td>Message Header</td><td>Uplink Message Header</td><td>4</td><td></td></tr>
<tr><td>Absolute Timestamp</td><td>
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="8">UNIX Timestamp MSB</td></tr>
<tr><td>2</td><td colspan="8">UNIX Timestamp</td></tr>
<tr><td>3</td><td colspan="8">UNIX Timestamp</td></tr>
<tr><td>4</td><td colspan="8">UNIX Timestamp LSB</td></tr>
</table>
</td><td>4</td><td>Absolute timestamp of the oldest event as base for recalculate if the relative timestamps of the following event objects</td></tr>
<tr><td>Analog value object (oldest in queue)</td><td>
<table>
<tr><td>Byte\bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>4</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td colspan="4">ObjectType</td><td colspan="4">ObjectNo</td></tr>
<tr><td></td><td colspan="8">0x51</td></tr>
<tr><td>2</td><td colspan="4">COT</td><td colspan="4">Status</td></tr>
</table>
</td><td>4</td><td>TimeStampOffset will be 0, because it's similar to the succeeding absolute time</td></tr>
</table>

	<table><tr><td></td><td>CYC</td><td>INT</td><td>EVT</td><td>LIM</td><td>IV</td><td>OV</td><td>LIM2</td><td>LIM2</td></tr><tr><td>3</td><td colspan="8">0</td></tr><tr><td>4</td><td colspan="8">Analogue value MSB</td></tr><tr><td>5</td><td colspan="8">Analogue value LSB</td></tr></table>COT: Cause of transmission - CYC: Cyclic event - INT: Interrogation triggered by button, downlink or first transmission after restart or re-join - EVT: Event, e.g. Delta Event - LIM: Limit value over/underrun Status: - IV: invalid, below measuring range - OV: overflow, measuring range exceeded - LIM1: Limit value1 violated - LIM2: Limit value2 violated		
Analog value object (oldest in queue)	See above								4	More Events in message if more in the event queue TimestampOffset is related to Timestamp

The analogue value objects must be included in the data packet in chronological order, not sorted by object number. An analogue value (equal to object number) can also be contained several times in a data packet, e.g. if events occur faster than can be transmitted.

Example telegram:

Base64: EQMEAWPQM6dVgB7jAABWiQAAAA=

Hex: 1103040163d033a755801ee30000568900000000

Decoded Payload(JSON):


```
{
  "data" : {
   "decoder" : {
    "info" : "comtac Cluey",
    "version" : "00.07"
   },
   "payload" : {
    "data" : {
     "analogueInputs" : [ {
      "cot" : {
       "cyclic" : true,
       "event" : false,
       "interrogation" : false,
       "limit" : false
      },
      "info" : {
       "id" : 5,
       "type" : "analogueValue"
      },
      "status" : {
       "invalid" : false,
       "limit1" : false,
       "limit2" : false,
       "overflow" : false
      },
      "timestamp" : {
       "string" : "Tue, 24 Jan 2023 19:38:15 GMT",
       "unix" : 1674589095
```

```
      },
      "value" : 7907
     }, {
      "cot" : {
       "cyclic" : true,
       "event" : false,
       "interrogation" : false,
       "limit" : false
      },
      "info" : {
       "id" : 6,
       "type" : "analogueValue"
      },
      "status" : {
       "invalid" : true,
       "limit1" : true,
       "limit2" : false,
       "overflow" : false
      },
      "timestamp" : {
       "string" : "Tue, 24 Jan 2023 19:38:15 GMT",
       "unix" : 1674589095
      },
      "value" : 0
     } ]
    },
    "device" : {
     "batteryLevel" : 0,
     "deviceStatus" : {
      "batteryPowered" : false,
      "bufferOverflow" : false,
      "configurationError" : false,
      "confirmationTimeout" : false,
      "deviceRestarted" : false,
      "lowSupplyVoltage" : false,
      "timeSynced" : true,
      "txCreditsConsumed" : false
     },
     "info" : {
      "deviceDesignation" : "Cluey-AM",
      "deviceId" : 17,
      "deviceManufacturer" : "comtac AG",
      "deviceVersion" : 3
     }
    }
   },
   "payloadLength" : 20,
   "port" : 23,
   "portFunction" : "AI_DATA"
  },
  "warnings" : [ ]
}
```


13.2.6 **Config data packet (uplink port 100)**

The Config data packet contains the parameter settings in the configuration file "cfg.txt" of the Cluey.

This data packet is only sent on request by an "APP Configuration Request" downlink data packet and contains the parameters requested in the downlink packet. If more parameters are requested than fit into one data packet, several data packets are transmitted consecutively.

For the assignment, the individual parameter objects are marked with a ParameterId and the length specification ParameterSize, which indicates the number of the following bytes that the parameter occupies in the payload.

Note that ParameterSize specifies the number of bytes of the parameter. Since a parameter can contain several values, individual values must be decoded using the size specification of the parameter values.

Sending measured value data packets always has higher priority than Config data packets.

Content	Structure	Size [bytes]	Remark
Message Header	Uplink Message Header	4	
ParameterID 1	Byte \Bit: 7 6 5 4 3 2 1 0 1: ParameterID	1	ID of the parameter
Length 1	Byte \Bit: 7 6 5 4 3 2 1 0 1: ParameterSize	1	Number of bytes of the following parameter
Parameter 1	Byte \Bit: 7 6 5 4 3 2 1 0 1: Parameter MSB ... N: Parameter LSB	1..N	Number of bytes corresponds to the specification in the preceding length byte
ParameterID 2	Byte \Bit: 7 6 5 4 3 2 1 0 1: ParameterID	1	ID of the parameter
Length 2	Byte \Bit: 7 6 5 4 3 2 1 0 1: ParameterLength	1	Number of bytes of the following parameter
Parameter 2	Byte \Bit: 7 6 5 4 3 2 1 0 1: Parameter MSB ... N: Parameter LSB	1-..N	Number of bytes corresponds to the specification in the preceding length byte
...	...	...	

13.2.6.1 Parameter IDs

The following parameter ID's are used to identify configuration parameters.

The following table shows the size of the respective parameter value and whether it is a parameter array, i.e. a selection mask must be used for addressing when changing a parameter value.

Parameter	Description	Value Size in bytes	Selection mask	ParameterID
Label	String with max. 30 chars	30		0x00
DefaultSupplyMode	0: external DC supply, 1: battery	1		0x01
BufferedOperation	0: not enabled, 1: enabled	1		0x02
BufferedOperationSpan	0..65535 s	2		0x03
PayloadFormat	0: static payload, 1: dynamic payload	1		0x04
MeasIntervalDcSupply	30..32000 ms	2		0x06
MeasIntervalBattery	30..32000 ms	2		0x07
TimeSyncInterval	1..65535 h	2		0x08
RejoinInterval	0..65535 d	2		0x09
MeasIntervalAi	1...65535 s	2		0x0A
IS_Enable	0: not enabled, 1: enabled	2	☒	0x0C
IS_Active	0: not inverted 1: inverted	2	☒	0x0D
IS_Invert	0: not enabled, 1: enabled	2	☒	0x0E
IS_DelayEnable	0: not enabled, 1: enabled	2	☒	0x0F
IS_DelayRising	0..65535 ms	2	☒	0x10
IS_DelayFalling	0..65535 ms	2	☒	0x11
IS_DelayScaling	0: ms, 1: s, 2: min	1	☒	0x12
IS_WiperEnable	0: not enabled, 1: enabled	2	☒	0x13
IS_WiperConfirmationTimeout	0..65535 s	2	☒	0x14
IS_DeflutterEnable	0: not enabled, 1: enabled	2	☒	0x15
IS_DeflutterInterval	0..65535 ms	2	☒	0x16
IS_DeflutterCount	0..65535	2	☒	0x17
IS_DoubleEnable	0: not enabled, 1: enabled	1	☒	0x18

IS_DoubleIntermediateStateTimoutEnable	0: not enabled, 1: enabled	1	☒	0x19
IS_DoubleIntermediateStateTimout	0..65535 s	2	☒	0x1A
IS_CounterEnable	0: not enabled, 1: enabled	2	☒	0x1B
IS_CounterMode	0: pulse mode, 1: operating time mode	2	☒	0x1D
IS_CounterScaling	0: ms, 1: s, 2: min, 3: h	1	☒	0x1E
AIS_Enable	0: not enabled, 1: 0-10V, 2: 0-20mA	1	☒	0x21
AIS_DeltaEnable	0: not enabled, 1: enabled)	1	☒	0x22
AIS_DeltaValue	0..65535	2	☒	0x23
AIS_Limit1Enable	0: not enabled, 1: enabled	1	☒	0x24
AIS_Limit2Enable	0: not enabled, 1: enabled	1	☒	0x25
AIS_Limit1DelayEnable	0: not enabled, 1: enabled	1	☒	0x26
AIS_Limit2DelayEnable	0: not enabled, 1: enabled	1	☒	0x27
AIS_LimitDelayScaling	(0: ms, 1: s, 2: min, 3: h	1	☒	0x28
AIS_Limit1DelayRising	0..65535	2	☒	0x29
AIS_Limit2DelayRising	0..65535	2	☒	0x2A
AIS_Limit1DelayFalling	0..65535	2	☒	0x2B
AIS_Limit2DelayFalling	0..65535	2	☒	0x2C
AIS_Limit1Value	0..65535	2	☒	0x2D
AIS_Limit2Value	0..65535	2	☒	0x2E
AIS_Limit1Hysteresis	0..65535	2	☒	0x2F
AIS_Limit2Hysteresis	0..65535	2	☒	0x30
AIS_Limit1Direction	0: lower, 1: upper	1	☒	0x31
AIS_Limit2Direction	0: lower, 1: upper	1	☒	0x32
ES_RisingEnable	0: not enabled, 1: enabled	2	☒	0x33
ES_FallingEnable	0: not enabled, 1: enabled	2	☒	0x34
ES_BlockedChangedEnable	0: not enabled, 1: enabled	2	☒	0x35
ES_CyclicDiEnable	0: not enabled, 1: enabled	2	☒	0x3F
ES_DiConfirmed	0: not enabled, 1: enabled	1	☒	0x40
ES_CyclicDiInterval	0..65535 s	2	☒	0x41
ES_Priority	0: low priority, 1: high priority	2	☒	0x36
ES_Delay	0..65535 ms	2	☒	0x37
ES_CyclicCntEnable	0: not enabled, 1: enabled	1	☒	0x45
ES_CntConfirmed	0: not enabled, 1: enabled	1	☒	0x46
ES_CyclicCntTimeDateWeekDaySel	0: Date, 1: Weekday	1	☒	0x47
ES_CyclicCntTimeDateWeekDay	0..7, 0: every day, 1: Monday,... OR 0..31, 0: every day, 1: 1st of month,...	1	☒	0x48
ES_CyclicCntTimeHour	0..23	1	☒	0x49
ES_CyclicCntTimeMinute	0..23	1	☒	0x4A
ES_CyclicCntTimeInterval	0: mask for above values, 0..65535 min	2	☒	0x4B
AES_Limit1RisingEnable		1	☒	0x38
AES_Limit2RisingEnable		1	☒	0x39
AES_Limit1FallingEnable		1	☒	0x3A
AES_Limit2FallingEnable		1	☒	0x3B
AES_DeltaEnable		1	☒	0x3C
AES_InvalidValueEnable		1	☒	0x3D
AES_OverflowValueEnable		1	☒	0x3E
ACS_Enable	(0: not enabled, 1: enabled	1	☒	0x4C
ACS_AlarmDelay	0..65535 ms	2	☒	0x4D
ACS_MotionDetectorSel	0: not enabled, 1...8: channel number	1	☒	0x4E

ACS_KeySwitchSel	0: not enabled, 1...8: channel number	1	☒	0x4F
ACS_DoorContactSel	0: not enabled, 1...8: channel number	1	☒	0x50
OS_TriggerEnable	1..65535 s	1	☒	0x5D
OS_TriggerPeriod	1..65535	2	☒	0x5E
OS_TriggerPeriodScaling	0: s, 1: min, 2: h	1	☒	0x5F
OS_Enable	0: not enabled, 1: enabled	2	☒	0x2D
OS_Invert	0: not enabled, 1: enabled	2	☒	0x2E
OS_Mode	0: static, 1: wiper	2	☒	0x2F
OS_WiperTime	0..65535 ms	2	☒	0x30
ConfigRequest	Downlink only			0xFE
Error	Error marker			0xFF

13.2.7 Info data packet (uplink port 101)

This data packet is sent on request by the

Info request downlink data packet sent.

or when the device status changes, e.g. when the power supply is changed from battery operation to external supply or vice versa.

It contains the device information on type and version as well as the current status and corresponds to the file header of the uplink packages.

Uplink Message Header

13.3 Downlink Messages

The Cluey can process different downlink messages. It can be used to request current data, control outputs and request or set parameters.

13.3.1 General interrogation of digital messages (downlink port 20)

This downlink message can be used to request the current status of the digital inputs. All digital messages together or selected ones can be requested. The messages are selected by setting the corresponding bit (0..15, corresponds to message 1...16) in the interrogation mask.

The response telegram only contains digital messages that are also activated in the configuration.

The general interrogation message is sent with a

Dynamic digital message data packet (uplink port 20)

or with a

Static data packet (uplink port 3)

answered, depending on the

Payload format setting.

In this case, the digital values are marked with the transmission cause COT= Interrogation (INT).

Content	Structure	Size [bytes]	Remark
Command	Byte \Bit: 7 6 5 4 3 2 1 0 1: 0x20	1	
Interrogation Mask	Byte \Bit: 7 6 5 4 3 2 1 0 1: Interrogation Mask MSB 2: Interrogation Mask LSB	2	

Example:

Port	**Data**		**Result**
20	Hex	20 00 01	Requesting the digital message 1
	Base64	IAAB	
20	Hex	20 00 04	Requesting the digital message 3
	Base64	IAAE	
20	Hex	20 FF FF	Request all digital messages *
	Base64	IP//	

*Reply telegram example

Base64: EQMEAWPQNmIRQAAAEkAAABNAAAA=

Hex: 1103040163d0366211400000124000001340000 0

Decoded payload:


```
{
  "data" : {
   "decoder" : {
    "info" : "comtac Cluey",
    "version" : "00.07"
   },
   "payload" : {
    "data" : {
     "digitalInputs" : [ {
      "cot" : {
       "cyclic" : false,
       "event" : false,
       "interrogation" : true,
       "limit" : false
      },
      "info" : {
       "id" : 1,
       "type : singlePointInfo
      },
      "status" : {
       "blocked" : false,
       "state" : 0
      },
      "timestamp" : {
       "string" : "Tue, 24 Jan 2023 19:49:54 GMT",
       "unix" : 1674589794
      }
     }, {
      "cot" : {
       "cyclic" : false,
       "event" : false,
       "interrogation" : true,
       "limit" : false
      },
      "info" : {
       "id" : 2,
       "type : singlePointInfo
```

```
      },
      "status" : {
        "blocked" : false,
        "state" : 0
      },
      "timestamp" : {
        "string" : "Tue, 24 Jan 2023 19:49:54 GMT",
        "unix" : 1674589794
      }
    }, {
      "cot" : {
        "cyclic" : false,
        "event" : false,
        "interrogation" : true,
        "limit" : false
      },
      "info" : {
        "id" : 3,
        "type : singlePointInfo
      },
      "status" : {
        "blocked" : false,
        "state" : 0
      },
      "timestamp" : {
        "string" : "Tue, 24 Jan 2023 19:49:54 GMT",
        "unix" : 1674589794
      }
    } ]
  },
  "device" : {
    "batteryLevel" : 0,
    "deviceStatus" : {
      "batteryPowered" : false,
      "bufferOverflow" : false,
      "configurationError" : false,
      "confirmationTimeout" : false,
      "deviceRestarted" : false,
      "lowSupplyVoltage" : false,
      "timeSynced" : true,
      "txCreditsConsumed" : false
    },
    "info" : {
      "deviceDesignation" : "Cluey-AM",
      "deviceId" : 17,
      "deviceManufacturer" : "comtac AG",
      "deviceVersion" : 3
    }
  }
 },
 "payloadLength" : 20,
 "port" : 20,
 "portFunction" : "DI_DATA"
},
"warnings" : [ ]
}
```


13.3.2 **Wiper acknowledgement (downlink port 20)**

The downlink message is used to acknowledge received wiper messages.


Wiper messages

This resets the self-holding of the wiper message in the Cluey.

Single or multiple wiper messages can be confirmed by setting the wiper message corresponding bit 0...15 (corresponds to message 1 to 16) of the wiper confirmation mask.

Content	Structure									Size [bytes]	Remark
Command	Byte \Bit	7	6	5	4	3	2	1	0	1	
	1	0x10									
Wiper Confirmation Mask	Byte \Bit	7	6	5	4	3	2	1	0	2	
	1	Wiper Confirmation Mask MSB									
	2	Wiper Confirmation Mask LSB									

Examples:

Port	Data		Result
20	Hex	10 00 01	Confirms wiper message 1
	Base64	EAAB	
20	Hex	10 00 04	Confirms wiper message 3
	Base64	EAAE	
20	Hex	10 FF FF	Confirms all wiper messages
	Base64	EP//	

13.3.3 General counter query (downlink port 21)

This downlink message can be used to request the current meter readings. All or selected counter values can be requested. The counter values are selected by setting the corresponding bit (0..15, corresponds to message 1...16) in the interrogation mask. Only the counter values that are activated are delivered.

The general interrogation message is sent with a

Dynamic count data packet (uplink port 20)

or with a

Static data packet (uplink port 3)

answered, depending on the

Payload format setting.

In this case, the count values are marked with the transmission cause COT= Interrogation(INT).

Content	Structure									Size [bytes]	Remark
Command	Byte \Bit	7	6	5	4	3	2	1	0	1	
	1	0x20									
Interrogation Mask	Byte \Bit	7	6	5	4	3	2	1	0	2	
	1	Interogation Mask MSB									
	2	Interogation Mask LSB									

Example:

Port	Data		Result
21	Hex	20 00 01	Requesting the count value 1
	Base64	IAAB	
21	Hex	20 00 04	Requesting the count value 3

	Base64	IAAE	
21	Hex	20 FF FF	Request all count values *
	Base64	IP//	

*Reply telegram example

Base64: EQMEAQNj0Dd2QUAAARFCQAABEUNAAAER

Hex: 110304010363d037764140000111424000011143400000111

Decoded payload:


```
{
  "data" : {
   "decoder" : {
    "info" : "comtac Cluey",
    "version" : "00.07"
   },
   "payload" : {
    "data" : {
     "counters" : [ {
      "cot" : {
       "cyclic" : false,
       "event" : false,
       "interrogation" : true,
       "limit" : false
      },
      "info" : {
       "id" : 1,
       "type" : "counter"
      },
      "status" : {
       "limit" : false,
       "overflow" : false,
       "reset" : false
      },
      "timestamp" : {
       "string" : "Tue, 24 Jan 2023 19:54:30 GMT",
       "unix" : 1674590070
      },
      "value : 273
     }, {
      "cot" : {
       "cyclic" : false,
       "event" : false,
       "interrogation" : true,
       "limit" : false
      },
      "info" : {
       "id" : 2,
       "type" : "counter"
      },
      "status" : {
       "limit" : false,
       "overflow" : false,
       "reset" : false
      },
      "timestamp" : {
       "string" : "Tue, 24 Jan 2023 19:54:30 GMT",
       "unix" : 1674590070
      },
      "value : 273
     }, {
      "cot" : {
       "cyclic" : false,
       "event" : false,
       "interrogation" : true,
       "limit" : false
      },
```

```
      "info" : {
        "id" : 3,
        "type" : "counter"
      },
      "status" : {
        "limit" : false,
        "overflow" : false,
        "reset" : false
      },
      "timestamp" : {
        "string" : "Tue, 24 Jan 2023 19:54:30 GMT",
        "unix" : 1674590070
      },
      "value : 273
     } ]
    },
    "device" : {
     "batteryLevel" : 0,
     "deviceStatus" : {
      "batteryPowered" : false,
      "bufferOverflow" : false,
      "configurationError" : false,
      "confirmationTimeout" : false,
      "deviceRestarted" : false,
      "lowSupplyVoltage" : false,
      "timeSynced" : true,
      "txCreditsConsumed" : false
     },
     "info" : {
      "deviceDesignation" : "Cluey-AM",
      "deviceId" : 17,
      "deviceManufacturer" : "comtac AG",
      "deviceVersion" : 3
     }
    }
   },
   "payloadLength" : 24,
   "port" : 21,
   "portFunction" : "CNT_DATA"
  },
  "warnings" : [ ]
}
```


13.3.4 **Reset counter (downlink port 21)**

This downlink message allows the counter readings to be reset to zero. All or selected counter values can be reset. The counter values are selected by setting the corresponding bit (0..15, corresponds to message 1...16) in the reset mask.

Dynamic counter value data packet (uplink port 21)

or with a

Static data packet (uplink port 3)

answered, depending on the

Payload format setting.

In the next transmission, the reset counter readings in the status with reset bit RES=1 set and in the transmission cause COT, the interrogation bit is set.

Content	Structure									Size [bytes]	Remark
Command	Byte \Bit	7	6	5	4	3	2	1	0	1	
	1	0x10									
Reset Mask	Byte \Bit	7	6	5	4	3	2	1	0	2	
	1	Reset Mask MSB									
	2	Reset Mask LSB									

Example:

Port	**Data**		**Result**
21	Hex	10 00 01	Requesting the count value 1
	Base64	EAAB	
21	Hex	10 00 04	Requesting the count value 3
	Base64	EAAE	
21	Hex	10 FF FF	Request all count values *
	Base64	EP//	

*Reply telegram example

Base64: EQIEAQRjzUt8QUAAAMlCQAAAyUNAAADJREAAAMk=

Hex: 110204010463cd4b7c41400000c942400000c943400000c944400000c9

Decoded payload:


```
{
  "data" : {
   "decoder" : {
    "info" : "comtac Cluey",
    "version" : "00.07"
   },
   "payload" : {
    "data" : {
     "counters" : [ {
      "cot" : {
       "cyclic" : false,
       "event" : false,
       "interrogation" : true,
       "limit" : false
      },
      "info" : {
       "id" : 1,
       "type" : "counter"
      },
      "status" : {
       "limit" : false,
       "overflow" : false,
       "reset" : true
      },
      "timestamp" : {
       "string" : "Tue, 24 Jan 2023 19:59:05 GMT",
       "unix" : 1674590345
      },
      "value" : 0
     }, {
      "cot" : {
       "cyclic" : false,
       "event" : false,
       "interrogation" : true,
       "limit" : false
      },
      "info" : {
```

```
        "id" : 2,
        "type" : "counter"
       },
       "status" : {
        "limit" : false,
        "overflow" : false,
        "reset" : true
       },
       "timestamp" : {
        "string" : "Tue, 24 Jan 2023 19:59:05 GMT",
        "unix" : 1674590345
       },
       "value" : 0
      }, {
       "cot" : {
        "cyclic" : false,
        "event" : false,
        "interrogation" : true,
        "limit" : false
       },
       "info" : {
        "id" : 3,
        "type" : "counter"
       },
       "status" : {
        "limit" : false,
        "overflow" : false,
        "reset" : true
       },
       "timestamp" : {
        "string" : "Tue, 24 Jan 2023 19:59:05 GMT",
        "unix" : 1674590345
       },
       "value" : 0
      } ]
     },
     "device" : {
      "batteryLevel" : 0,
      "deviceStatus" : {
       "batteryPowered" : false,
       "bufferOverflow" : false,
       "configurationError" : false,
       "confirmationTimeout" : false,
       "deviceRestarted" : false,
       "lowSupplyVoltage" : false,
       "timeSynced" : true,
       "txCreditsConsumed" : false
      },
      "info" : {
       "deviceDesignation" : "Cluey-AM",
       "deviceId" : 17,
       "deviceManufacturer" : "comtac AG",
       "deviceVersion" : 3
      }
     }
    },
    "payloadLength" : 24,
    "port" : 21,
    "portFunction" : "CNT_DATA"
   },
   "warnings" : [ ]
  }
}
```


13.3.5 General interrogation of analogue values (downlink port 23)

With this downlink message, the current analogue values can be requested. All or selected analogue values can be requested. The analogue values are selected by setting the corresponding bit (0..15, corresponds to message 1...16) in the interrogation mask. Only the analogue values that are activated are delivered.

The general interrogation message is sent with a

Dynamic analogue value data packet (uplink port 23)

or with a

Static data packet (uplink port 3)

answered, depending on the

Payload format setting.

In this case, the analogue values are marked with the transmission cause COT= Interrogation(INT).

Content	Structure									Size [bytes]	Remark
Command	Byte \Bit	7	6	5	4	3	2	1	0	1	
	1	0x20									
Interrogation Mask	Byte \Bit	7	6	5	4	3	2	1	0	2	
	1	Interrogation Mask MSB									
	2	Interrogation Mask LSB									

Example:

Port	**Data**		**Result**
23	Hex	20 00 01	Request analogue value 1
	Base64	IAAB	
23	Hex	20 00 04	Requesting the analogue value 3
	Base64	IAAE	
23	Hex	20 FF FF	Request all analogue values *
	Base64	IP//	

*Reply telegram example

Base64: EQMEAWPQOVpUSAAABVQB7wAABWSQAAAA=

Hex: 1103040163d0395a54480000000055401ef00000564900000000

Decoded payload:


```
{
  "data" : {
   "decoder" : {
    "info" : "comtac Cluey",
    "version" : "00.07"
   },
   "payload" : {
    "data" : {
     "analogInputs" : [ {
      "cot" : {
       "cyclic" : false,
       "event" : false,
       "interrogation" : true,
       "limit" : false
      },
      "info" : {
       "id" : 4,
       "type : analogueValue
      },
      "status" : {
```

```
      "invalid" : true,
      "limit1" : false,
      "limit2" : false,
      "overflow" : false
     },
     "timestamp" : {
      "string" : "Tue, 24 Jan 2023 20:02:34 GMT",
      "unix" : 1674590554
     },
     "value" : 0
    }, {
     "cot" : {
      "cyclic" : false,
      "event" : false,
      "interrogation" : true,
      "limit" : false
     },
     "info" : {
      "id" : 5,
      "type : analogueValue
     },
     "status" : {
      "invalid" : false,
      "limit1" : false,
      "limit2" : false,
      "overflow" : false
     },
     "timestamp" : {
      "string" : "Tue, 24 Jan 2023 20:02:34 GMT",
      "unix" : 1674590554
     },
     "value" : 7920
    }, {
     "cot" : {
      "cyclic" : false,
      "event" : false,
      "interrogation" : true,
      "limit" : false
     },
     "info" : {
      "id" : 6,
      "type : analogueValue
     },
     "status" : {
      "invalid" : true,
      "limit1" : true,
      "limit2" : false,
      "overflow" : false
     },
     "timestamp" : {
      "string" : "Tue, 24 Jan 2023 20:02:34 GMT",
      "unix" : 1674590554
     },
     "value" : 0
    } ]
   },
   "device" : {
    "batteryLevel" : 0,
    "deviceStatus" : {
     "batteryPowered" : false,
     "bufferOverflow" : false,
     "configurationError" : false,
     "confirmationTimeout" : false,
     "deviceRestarted" : false,
     "lowSupplyVoltage" : false,
     "timeSynced" : true,
     "txCreditsConsumed" : false
    },
    "info" : {
     "deviceDesignation" : "Cluey-AM",
     "deviceId" : 17,
     "deviceManufacturer" : "comtac AG",
```

```
        "deviceVersion" : 3
       }
      }
     },
    "payloadLength" : 26,
    "port" : 23,
    "portFunction" : "AI_DATA"
   },
   "warnings" : [ ]
}
```


13.3.6 Control outputs (downlink port 20)

These downlink messages can be used to control the digital outputs of the Cluey. Individual or several outputs can be controlled with one message. The message does not trigger an uplink telegram.

Only outputs activated by the parameter

OS_Enable

are activated.

The state of an output can be read back when the associated digital input is activated and operated in active mode. If the change event for the input is activated, the state of the output when it changes is sent with a corresponding digital message telegram. Likewise, if the message transmission is cyclic or by query, the current state of the output is sent in the message telegram.

13.3.6.1 Statically control outputs (downlink port 20)

The downlink message allows static setting or resetting of the digital outputs. Static control is only possible if the

OS_Mode

is set to static in the configuration.

One or more outputs can be controlled. Only the outputs that are selected with the Select Mask are controlled. A bit which has been set in the Select Mask at the bit position corresponding to the output, selects the output. A 1 or 0 at the bit position corresponding to an output determines whether the output is switched on or off.

Please note that the behaviour of the output may be inverted in relation to the control if the inversion is compensated by the parameter

OS_Invert

who is active.

Content	Structure									Size [bytes]	Remark
Command	Byte \Bit	7	6	5	4	3	2	1	0	1	
	1	0x32									
Set Value	Byte \Bit	7	6	5	4	3	2	1	0	2	1 at the bit position sets the corresponding output , 0 resets it
	1	Set Value MSB									
	2	Set Value LSB									
Select Mask	Byte \Bit	7	6	5	4	3	2	1	0	2	1 at the bit position selects
	1	Select Mask MSB									

	2	Select Mask LSB		the corresponding output for control. With 0, the output remains unchanged

Examples:

Port	Data		Result
20	Hex	32 FF FF 00 01	Switch on output 1
	Base64	Mv//AAE=	
20	Hex	32 FF FF 00 02	Switch on output 2
	Base64	Mv//AAI=	
20	Hex	32 FF FF 00 04	Switch on output 3
	Base64	Mv//AAQ=	
20	Hex	32 FF FF 00 08	Switch on output 4
	Base64	Mv//AAg=	
20	Hex	32 00 00 00 01	Switch off output 1
	Base64	MgAAAAE=	
20	Hex	32 00 00 00 02	Switch off output 2
	Base64	MgAAAAI=	
20	Hex	32 00 00 00 04	Switch off output 3
	Base64	MgAAAAQ=	
20	Hex	32 00 00 00 08	Switch off output 4
	Base64	MgAAAAg=	
20	Hex	32 00 01 00 05	Switch off output 3, switch on output 1
	Base64	MgABAAU=	
20	Hex	32 00 04 00 05	Switch on output 3, switch off output 1
	Base64	MgAEAAU=	

13.3.6.2 Output wiper (downlink port 20)

The downlink message can be output as wiper pulses via the digital outputs. The pulse duration can also be sent optionally, otherwise the pulse duration corresponds to the configured value in the configuration file.

Parameters: OS_WiperTime

The wiper control is only possible if the

OS_Mode

is set to wiper in the configuration.

A bit set at the output bit position in the select mask determines which output(s) is/are controlled.

13.3.6.3 Output wiper with configured duration (downlink port 20)

With the following downlink message, the wiper is output with the wipe duration set in the configuration file.

Content	Structure	Size [bytes]	Remark
Command	Byte \Bit: 7 6 5 4 3 2 1 0 1: 0x32	1	
Select Mask	Byte \Bit: 7 6 5 4 3 2 1 0 1: Select Mask MSB 2: Select Mask LSB	2	1 at the bit position selects the corresponding output for control. With 0, the output remains unchanged

Examples:

Port	Data		Result
20	Hex	33 00 01	Output wiper to output 1
	Base64	MwAB	
20	Hex	33 00 02	Output wiper to output 2
	Base64	MwAC	
20	Hex	33 00 04	Output wiper to output 3
	Base64	MwAE	
20	Hex	33 00 08	Output wiper to output 4
	Base64	MwAI	
20	Hex	33 00 05	Output wiper to output 1 and 3
	Base64	MwAF	

13.3.6.4 Output wiper with controlled duration (downlink port 20)

The wiper is output with the following downlink message, where the desired wipe duration is also sent in the message.

It should be noted that the wiper durations for all outputs must be transmitted.

Currently, only 4 outputs are available, therefore only 4 wipe durations must be transmitted. The wipe duration is given in milliseconds.

The wiper durations stored in the configuration file are not changed by the message.

Content	Structure	Size [bytes]	Remark
Command	Byte \Bit: 7 6 5 4 3 2 1 0 1: 0x33	1	
Select Mask	Byte \Bit: 7 6 5 4 3 2 1 0 1: Select Mask MSB 2: Select Mask LSB	2	1 at the bit position selects the corresponding output for control. With 0, the output remains unchanged
Wiper Duration Output 1	Byte \Bit: 7 6 5 4 3 2 1 0 1: Wiper Duration MSB 2: Wiper Duration LSB	2	Wipe duration for output 1

Wiper Duration Output 2	Byte \Bit	7	6	5	4	3	2	1	0	2	Wipe duration for output 2 (optional, if only output 1 is controlled)
	1	Wiper Duration MSB									
	2	Wiper Duration LSB									
Wiper Duration Output 3	Byte \Bit	7	6	5	4	3	2	1	0	2	Wipe duration for (optional, if only output 1 &2 is controlled)output 3
	1	Wiper Duration MSB									
	2	Wiper Duration LSB									
Wiper Duration Output 4	Byte \Bit	7	6	5	4	3	2	1	0	2	Wipe duration for output 4 (optional, if only output 1,2 & 3 is controlled)
	1	Wiper Duration MSB									
	2	Wiper Duration LSB									

Examples;
Wipe duration output 1: 500ms → 0x01f4.
Wipe duration output 2: 600ms → 0x0258.
Wipe duration output 3: 800ms → 0x0320.
Wipe duration output 4: 1s → 0x03E8.

Port	**Data**		**Result**
20	Hex	33 00 01 F4 (02 58 03 20 03 E8)	Output wiper with 500ms duration on output 1 (bytes in brackets are not needed)
	Base64	MwABAfQCWAMgA+g=	
20	Hex	33 00 02 01 F4 02 58 (03 20 03 E8)	Output wiper with 600ms duration on output 2 (bytes in brackets are not needed)
	Base64	MwACAfQCWAMgA+g=	
20	Hex	33 00 04 01 F4 02 58 03 20 (03 E8)	Output wiper with 800ms duration on output 3 (bytes in brackets are not needed)
	Base64	MwAEAfQCWAMgA+g=	
20	Hex	33 00 04 01 F4 02 58 03 20 03 E8	Output wiper with 1s duration on output 4
	Base64	MwAEAfQCWAMgA+g=	
20	Hex	33 00 05 01 F4 02 58 03 20 (03 E8)	Output wiper on output 1 (500ms) and 3(800ms)
	Base64	MwAFAfQCWAMgA+g=	

13.3.7 **Configuration parameter query (downlink port 100)**

The following downlink messages can be used to query the configuration data of the Cluey.

13.3.8 **Query all configuration parameters (downlink port 100)**

The following downlink message allows you to query all configuration parameters.

As this is very large data, the uplink response will usually be split into several data packets.

Content	Structure									Size [bytes]	Remark
Command	Byte \Bit	7	6	5	4	3	2	1	0	1	
	1	0xFE									
Parameter ID	Byte \Bit	7	6	5	4	3	2	1	0	1	
	1	0xFF									

Message:

Port	Data		Result
100	Hex	FE FF	Request all configuration parameters
	Base64	/v8=	

13.3.9 Querying individual configuration parameters (downlink port 100)

The following downlink message enables the query of individual, i.e. a single or several selected configuration parameters. The message can thus contain the parameter ID of one or more parameters. Accordingly, the response telegram contains the requested parameter values. The structure of the response telegram is described here:

If an invalid parameter ID is requested, the response telegram contains the error ID (0xFE).

Structure of the request telegram

Content	Structure									Size [bytes]	Remark
Command	Byte \Bit	7	6	5	4	3	2	1	0	1	
	1	0xFE									
Number of the following ID's	Byte \Bit	7	6	5	4	3	2	1	0	1	
	1	1....N									
Parameter ID 1	Byte \Bit	7	6	5	4	3	2	1	0	1	Parameter ID of the desired parameter
	1	Parameter ID									
Further parameter IDs	Byte \Bit	7	6	5	4	3	2	1	0	2..N	Optionally, further parameter IDs can follow
	1	Parameter ID									

13.3.9.1 Example : Query parameter "Label" (ID=0x00)

Content of the configuration file:
Label="Cluey AM " (max 30 chars)

Query message (downlink)

Port	Data		Result
100	Hex	FE 01 00	Query label, telegram contains an ID, ID -Label = 00
	Base64	/gEA	

Reply message(uplink):

Base64: EQMEAQAeQ0xVRVkgQU0gICAgICAgICAgICAgICAgICAgICAg

Hex: 11030401001e434c55455920414d2020202020202020202020202020202020202020202020

Structure

Header	ID	Bytes	Parameter (text)
11 03 04 01	00	1F	434C554559204B4D2054657374676572C3A4742050726F6A656B7420455744 ⇨ "Cluey AM "

Decoded payload:


```
"data" : {
 "decoder" : {
  "info" : "comtac Cluey",
  "version" : "00.07"
 },
 "payload" : {
  "data" : {
   "parameters" : [ {
    }, "name" : "LABEL",
    "value" : "Cluey AM "
   } ]
  },
  "device" : {
   "batteryLevel" : 0,
   "deviceStatus" : {
    "batteryPowered" : false,
    "bufferOverflow" : false,
    "configurationError" : false,
    "confirmationTimeout" : false,
    "deviceRestarted" : false,
    "lowSupplyVoltage" : false,
    "timeSynced" : true,
    "txCreditsConsumed" : false
   },
   "info" : {
    "deviceDesignation" : "Cluey-AM",
    "deviceId" : 17,
    "deviceManufacturer" : "comtac AG",
    "deviceVersion" : 3
   }
  }
 },
 "payloadLength" : 36,
 "port : 100,
 "portFunction" : "CONFIG"
},
"warnings" : [ ]
}
```


13.3.9.2 Example: Query parameter "IS_DelayRising" (ID=0x10)

Content of the configuration file:

IS_DelayRising=00100;00100;00100;00100;00100;00100;00100;00100 (0..65535)

Query message (downlink)

Port	Data		Result
100	Hex	FE 01 10	Query IS_DelayRising, telegram contains an ID, ID -Label = 0x10
	Base64	/gEQ	

Reply message(uplink):

Base64: EQMEARAQAGQAZABkAGQAZABkAGQAZA==

Hex: 11030401 10 10 0064 0064 0064 0064 0064 0064 0064 0064 0064

Structure:

Header	ID	Size	Value1	Value2	Value3	Value4	Value5	Value6	Value7	Value8
11 03 04 01	10	10 = 8 * 2 bytes	0064	0064	0064	0064	0064	0064	0064	0064

Decoded payload:


```
{
  "data" : {
   "decoder" : {
    "info" : "comtac Cluey",
    "version" : "00.07"
   },
   "payload" : {
    "data" : {
     "parameters" : [ {
      "name" : "IS_DELAY_RISING",
      "unit" : "ms",
      "values" : [ 100, 100, 100, 100, 100, 100, 100 ]
     } ]
    },
    "device" : {
     "batteryLevel" : 0,
     "deviceStatus" : {
      "batteryPowered" : false,
      "bufferOverflow" : false,
      "configurationError" : false,
      "confirmationTimeout" : false,
      "deviceRestarted" : false,
      "lowSupplyVoltage" : false,
      "timeSynced" : true,
      "txCreditsConsumed" : false
     },
     "info" : {
      "deviceDesignation" : "Cluey-AM",
      "deviceId" : 17,
      "deviceManufacturer" : "comtac AG",
      "deviceVersion" : 3
     }
    }
   },
   "payloadLength" : 22,
   "port : 100,
   "portFunction" : "CONFIG"
  },
  "warnings" : [ ]
 }
```


13.3.9.3 Example: Query parameter " IS_Enable" (ID=0x0C)

Content of the configuration file:

IS_Enable=1;1;1;1;0;0;0 (0: not enabled, 1: enabled)

Query message (downlink)

Port	Data		Result
100	Hex	FE 01 0C	Query IS_Enable, telegram contains an ID, ID = 0x0C
	Base64	/gEM	

Reply message(uplink):

Base64: EQMEAQwCAAc=

Hex: 11030401 0c 02 0007

Structure:

Header	ID	Size	Value = 0x0007 = 0b0000 0000 0111 = IS_Enable 16...1															
11 03 04 01	1c	02 = 2 Bytes= 16Bits	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

			=disabled	=disabled	=disabled	=disabled	=disabled	=disabled	=disabled	=disabled	=disabled	=disabled	=disabled	=disabled	=disabled	=disabled	enable d	enable d	enable d

Decoded payload:


```
{
   "data" : {
    "decoder" : {
     "info" : "comtac Cluey",
     "version" : "00.07"
    },
    "payload" : {
     "data" : {
      "parameters" : [ {
       "name" : "IS_ENABLE",
       "value" : "00000000000111"
      } ]
     },
     "device" : {
      "batteryLevel" : 0,
      "deviceStatus" : {
       "batteryPowered" : false,
       "bufferOverflow" : false,
       "configurationError" : false,
       "confirmationTimeout" : false,
       "deviceRestarted" : false,
       "lowSupplyVoltage" : false,
       "timeSynced" : true,
       "txCreditsConsumed" : false
      },
      "info" : {
       "deviceDesignation" : "Cluey-AM",
       "deviceId" : 17,
       "deviceManufacturer" : "comtac AG",
       "deviceVersion" : 3
      }
     }
    },
    "payloadLength" : 8,
    "port : 100,
    "portFunction" : "CONFIG"
   },
   "warnings" : [ ]
 }
```


13.3.9.4 Example: Query parameter " IS_CounterMode" (ID=0x1D)

Content of the configuration file:

IS_CounterMode=0;0;0;0;0;0;0 (0: pulse mode, 1: operating time mode)

Query message (downlink)

Port	Data		Result
100	Hex	FE 01 1D	Query IS_CounterMode, telegram contains an ID, ID -Label = 0x1D
	Base64	/gEd	

Reply message(uplink):

Base64: EQMEAR0CAAA=

Hex: 110304011d020000

Structure:

Header	ID	Size	Value = 0x0000 = 0b0000 0000 0000 = counter-Mode 16...1															
11 03 04 01	1D	02 = 2 Bytes= 16Bits	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Decoded payload:


```
{
 "applicationID" : "1",
{
   "data" : {
    "decoder" : {
     "info" : "comtac Cluey",
     "version" : "00.07"
    },
    "payload" : {
     "data" : {
      "parameters" : [ {
       "name" : "IS_COUNTER_MODE",
       "values" : [ "pulse", "pulse", "pulse", "pulse", "pulse", "pulse", "pulse", "pulse", "pulse", "pulse", "pulse", "pulse", "pulse", "pulse", "pulse" ]
      } ]
     },
     "device" : {
      "batteryLevel" : 0,
      "deviceStatus" : {
       "batteryPowered" : false,
       "bufferOverflow" : false,
       "configurationError" : false,
       "confirmationTimeout" : false,
       "deviceRestarted" : false,
       "lowSupplyVoltage" : false,
       "timeSynced" : true,
       "txCreditsConsumed" : false
      },
      "info" : {
       "deviceDesignation" : "Cluey-AM",
       "deviceId" : 17,
       "deviceManufacturer" : "comtac AG",
       "deviceVersion" : 3
      }
     }
    },
    "payloadLength" : 8,
    "port : 100,
    "portFunction" : "CONFIG"
  },
  "warnings" : [ ]
 }
```


13.3.9.5 Example: Query parameter " AIS_Enable" (ID=0x21)

Content of the configuration file:

AIS_Enable=0;0;0;2;2;0;0 (0: not enabled, 1: 0-10V, 2: 0-20mA)

Query message (downlink)

Port	**Data**		**Result**
100	Hex	FE 01 21	Query IS_DelayRising, telegram contains an ID, ID -Label = 0x21
	Base64	/gEh	

Reply message(uplink):

Base64: EQMEASEQAAAAAAAAIAAgACAAAA==

Hex: 11030401 21 10 0000 0000 0002 0002 0000 0000

Structure:

Header	ID	Size	Value1	Value2	Value3	Value4	Value5	Value6	Value7	Value8
11030401	21	10 = 8 * 2 bytes	0000 =disabled	0000 =disabled	0000 =disabled	0002 ="0-20mA"	0002 ="0-20mA"	0002 ="0-20mA"	0000 =disabled	0000 =disabled

Decoded payload:


```
{
   "data" : {
    "decoder" : {
      "info" : "comtac Cluey",
      "version" : "00.07"
    },
    "payload" : {
      "data" : {
        "parameters" : [ {
          }, "name" : "AIS_ENABLE",
          "values" : [ "disabled", "disabled", "disabled", "0-20mA", "0-20mA", "0-20mA", "disabled", "disabled" ]
        } ]
      },
      "device" : {
        "batteryLevel" : 0,
        "deviceStatus" : {
          "batteryPowered" : false,
          "bufferOverflow" : false,
          "configurationError" : false,
          "confirmationTimeout" : false,
          "deviceRestarted" : false,
          "lowSupplyVoltage" : false,
          "timeSynced" : true,
          "txCreditsConsumed" : false
        },
        "info" : {
          "deviceDesignation" : "Cluey-AM",
          "deviceId" : 17,
          "deviceManufacturer" : "comtac AG",
          "deviceVersion" : 3
        }
      }
    },
    "payloadLength" : 22,
    "port : 100,
    "portFunction" : "CONFIG"
  },
  "warnings" : [ ]
 }
```


13.3.10 **Change configuration parameters (downlink port 100)**

Parameters can also be changed via downlink message. In doing so, the values are changed permanently and are then also to be kept in the configuration file accordingly.

The structure of the downlink message is as follows.

It begins with the parameter ID that addresses the parameter to be changed. Followed by a length byte that contains the number of bytes that follow.

This is followed by one or more bytes for the parameter value. The number of bytes for the parameter value can be taken from the parameter table.

For configuration parameters that form an array of values (several comma-separated values per line in the configuration file), a bit mask following the parameter value must be used to select which of the array values (index) is to be changed.

The position of the bit set in the mask (0...15) determines the index of the parameter. The index 0 selects the first parameter of a line. If, although the parameter is an array, there is no selection mask, all values of the parameter are set to the same parameter value.

In this case, this must be taken into account accordingly in the length byte.

Structure of the request telegram

<table>
<tr><th>Content</th><th>Structure</th><th>Size [bytes]</th><th>Remark</th></tr>
<tr><td>Parameter ID</td><td><table><tr><td>Byte \Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>1</td><td colspan="8">0x00...0xFD</td></tr></table></td><td>1</td><td>See Parameter ID's</td></tr>
<tr><td>Number of the following bytes</td><td><table><tr><td>Byte \Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>1</td><td colspan="8">N + M</td></tr></table></td><td></td><td>Number of bytes (N) for the value, plus number of bytes (M=2) for the selection mask that may be necessary depending on the addressed parameter Mask</td></tr>
<tr><td>Value according to parameter</td><td><table><tr><td>Byte \Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>1</td><td colspan="8">MSB</td></tr><tr><td>...</td><td colspan="8"></td></tr><tr><td>N</td><td colspan="8">LSB</td></tr></table></td><td>1....N</td><td>Bytes Parameter value</td></tr>
<tr><td>Optional: Mask</td><td><table><tr><td>Byte \Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>1</td><td colspan="8">MSB (bit15...bit8) of the mask</td></tr><tr><td>2</td><td colspan="8">LSB (bit7...bit0) of the mask</td></tr></table></td><td></td><td>Selection mask determines the index of the parameter to be changed in a parameter array.</td></tr>
</table>

Explanatory examples

Port	**Data**	**Result**
100	03 02 00 0A	Set BufferedOperationTimeout to 10s
100	01 01 00	Set DefaultSupplyMode to external (default: battery) If no external power supply is present, device goes to sleep after 10s
100	02 01 00	Turn off BufferedOperation
100	04 01 00	Switch to static payload
100	05 01 07	Enable TS, DI and CNT in static payload
100	05 01 00	Disable TS, DI and CNT in static payload
100	07 02 00 1E	Set MeasInterval (battery supply) to 30ms
100	06 02 07 D0	Set MeasInterval (external supply) to 2000ms
100	08 02 00 01	Set TimeSyncInterval to 1h

100	0B 02 00 FF	Enable all inputs (8 out of possible 16)
100	0D 02 00 FF	Invert all inputs (8 out of possible 16)
100	0E 04 00 FF 00 01	Enable delay for the first input
100	25 02 00 01	Enable rising event on the first input, all other disabled
100	25 02 00 01 FE 01 25	Enable rising event on the first input and request new config after
100	2D 04 11 5C 00 04	Set AIS_Limit2 Value for analogue input 3 to 4444
100	2D 02 11 5C	Set AIS_Limit2Value for all analogue inputs to 4444

13.3.11 **Device Info Request (Downlink Port 101)**

The following down-link message can be used to request a device uplink telegram.

Content	Structure									Size [bytes]	Remark
Command	Byte \Bit	7	6	5	4	3	2	1	0	1	RS 1→ Triggers Device Info uplink
	1	0	0	0	0	0	0	0	1		

Port	**Data**		**Result**
101	Hex	01	Triggers Device Info uplink
	Base64	AQ==	

Example answer:

Base64: EQMEAQ==

Hex: 11030401

Decoded payload:


```
{
  "data" : {
   "decoder" : {
    "info" : "comtac Cluey",
    "version" : "00.07"
   },
   "payload" : {
    "device" : {
     "batteryLevel" : 0,
     "deviceStatus" : {
      "batteryPowered" : false,
      "bufferOverflow" : false,
      "configurationError" : false,
      "confirmationTimeout" : false,
      "deviceRestarted" : false,
      "lowSupplyVoltage" : false,
      "timeSynced" : true,
      "txCreditsConsumed" : false
     },
     "info" : {
      "deviceDesignation" : "Cluey-AM",
      "deviceId" : 17,
      "deviceManufacturer" : "comtac AG",
      "deviceVersion" : 3
     }
    }
   },
   "payloadLength" : 4,
   "port" : 101,
   "portFunction" : "INFO
  },
        "warnings" : [ ]
}
```

13.3.12 **Device Reset (Downlink Port 105)**

The Cluey can be restarted with the following downlink message. The Cluey behaves in the same way as a restart after voltage connection, i.e. if OTAA has been configured, a join is carried out and then either a static data telegram or dynamic telegrams with digital, counter and analogue values are sent, depending on the payload format setting.

Content	Structure									Size [bytes]	Remark
Command	Byte \Bit	7	6	5	4	3	2	1	0	1	Triggers Restart of device
	1	0	0	0	0	0	0	0	1		

Port	**Data**		**Result**
105	Hex	01	Unit reset
	Base64	AQ==	