



iotraffic

iotec GmbH

Albert-Einstein-Straße 30

49076 Osnabrück

0541 / 343 711-00

info@iotec.de

www.iotec-gmbh.de

Rev. 1.0

23. August 2024

1 Hinweise

Dies ist die Anleitung für die Nutzung des iottraffic Systems mit der Artikelnummer TR01-01 der iotec GmbH.

1.1 Sicherheitshinweise

Verändern Sie das System nur unter Berücksichtigung der in diesem Handbuch aufgeführten Hinweise und Anweisungen.

1.2 Urheberrechte

Dieses Dokument ist urheberrechtlich geschützt. Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Dokumentes oder Teilen daraus, sind der iotec GmbH vorbehalten. Geschrieben von Daniel Kümper im Auftrag der iotec GmbH aktualisiert am 16.10.2023.

2 Produktbeschreibung

Das iotraffic Messsystem dient zur permanenten Erfassung der Anzahl von Verkehrsteilnehmern, unterteilt in 7 Gruppen. Die Teilnehmer werden gezählt wenn sie eine definierte Linie, im durch die Kamera erfassten Bereich, überschreiten. Diese Daten werden in einem einstellbaren Sendeintervall aufgenommen, aggregiert und per Funkschnittstelle an eine LoRaWAN IoT-Plattform übermittelt. Die Energieversorgung erfolgt durch eine externe Spannungsversorgung mit 48V und mindestens 0,7A.



2.1 Vorbedingung zur Nutzung des iotraffic Systems

Für den Einsatz des iotraffic Systems muss der Anwender Zugriff zu einem LoRaWAN Portal (z.B. The Things Network, Niotix oder Zenner) haben. Zur Einrichtung des iotraffic Systems werden die mitgelieferten Anmeldeinformationen benötigt:

- LoRaWAND DEV-EUI
- LoRaWAN App-Key
- WiFi Name (SSID)
- WiFi Passwort (WPA2)
- Benutzerpasswort des Tunnelbenutzers (iotraffic5)

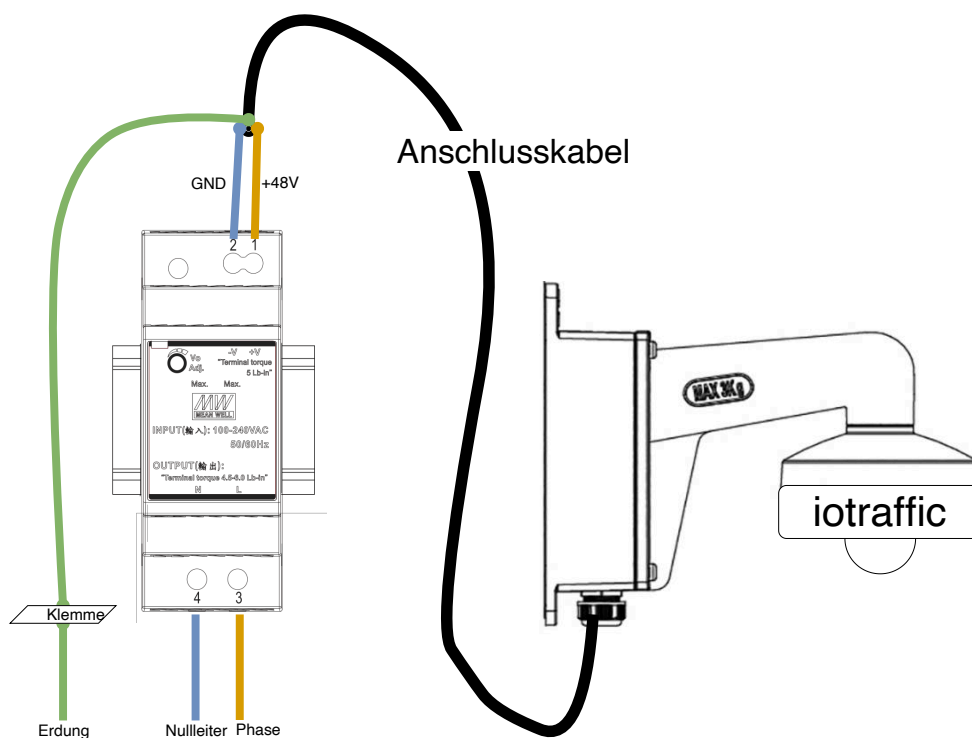
3 Konfiguration des iotraffic Systems am Standort

3.1 Portal Registrierung

Das iotraffic System muss zunächst in einem LoRaWAN Portal mit den mitgelieferten Zugangsdaten registriert werden. Hierzu sind die korrekte DEV-EUI und der App-Key bei Auswahl des Join-Verfahrens **OTAA** einzutragen. Desweiteren ist es wichtig iotraffic als **Class-C** Device anzulegen, damit zu jeder Zeit eine bidirektionale Kommunikation möglich ist.

3.2 Elektrischer Anschluss

Die folgende Abbildung zeigt den Anschluss des Netzteils und des iotraffic Systems. Die Erdung kann mit einer Klemme Durchverbunden werden.



Elektrischer Anschluss des iotraffic Systems

- Schraubterminal 1: braune Ader des iotraffic Systems (+48V)
- Schraubterminal 2: blaue Ader des iotraffic Anschlusskabels (Masse)
- Schraubterminal 3: Phase der 230 V Installation
- Schraubterminal 4: Nulleiter der 230V Installation

Nachdem das iotraffic System im LoRaWAN Portal registriert wurde, physisch installiert und elektrisch angeschlossen ist, sind folgende Schritte zu unternehmen.

3.3 Aktivieren des WiFi Moduls über das niotix

Die Konfiguration der Zähllinien erfolgt über die integrierte WiFi Verbindung. Diese ist jedoch im Normalzustand des Systems deaktiviert. Nach dem Join-Prozess des iottraffic Systems sollte im LoRaWAN Portal das entsprechende Gerät, z.B. 2310TR007, ausgewählt und der Downlink toggleWiFion gesendet werden.

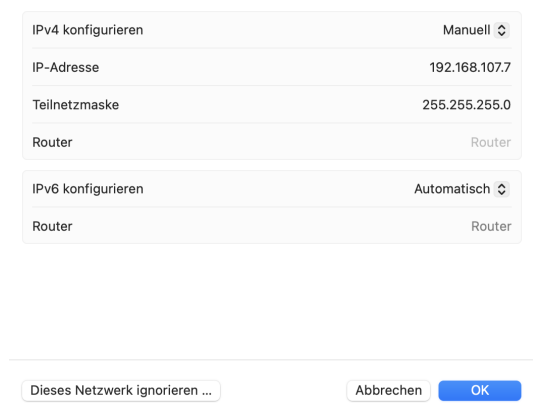
Wichtig: Nach der Konfiguration ist das Wifi-Modul wieder durch den Downlink toggleWiFi0ff zu deaktivieren. Ein Reboot oder Reset führt ebenfalls dazu, dass das WiFi deaktiviert ist.



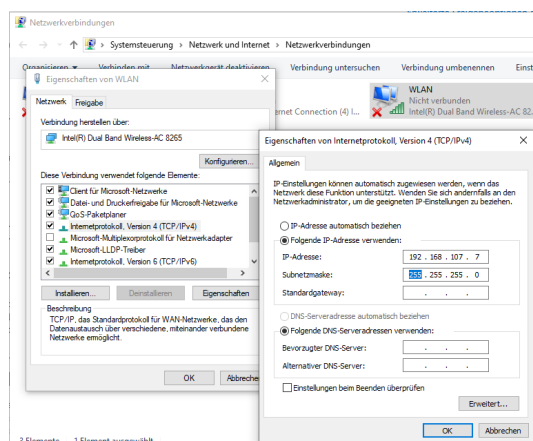
Beispielabbildung Downlink im Niotix Portal.

Nachdem der Downlink gesendet wurde ist das iottraffic System unter dem mitgelieferten WiFi Namen (SSID) verfügbar. Ferner wird das mitgelieferte Passwort benötigt um sich zu verbinden. Das iottraffic System hat eine statische IP Konfiguration auf der WiFi Schnittstelle. Die IP Adresse des iottraffic Systems lautet: 192.168.107.1. Somit muss ein zu-verbindender PC eine statische WiFi Konfiguration im gleichen Subnetz haben (IP Adresse darf hier nicht 192.168.107.1 sein). Beispielkonfiguration für einen PC:

- IP Adresse: 192.168.107.7
- Subnetzmaske 255.255.255.0
- Gateway und Nameserver sind nicht relevant.



Beispielkonfiguration WiFi für Mac.



Beispielkonfiguration WiFi für Windows.

Verbinden Sie sich mit dem WiFi und passen die IP-Konfiguration wie oben beschrieben an.

3.4 Verbindung des Benutzer Endgerätes mit dem iotraffic System

Zur Verbindung mit der Oberfläche des iotraffic Systems ist ein SSH-Tunnel aufzubauen. Dies kann wahlweise mit dem SSH-Client des Betriebssystems oder der GUI-basierten Software Putty \ <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> erfolgen.

3.4.1 SSH-Client des Betriebssystems (Win, Mac, Linux)

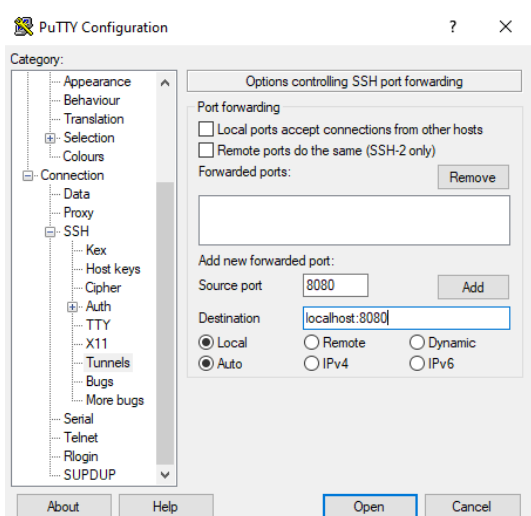
Zur Verbindung mit dem SSH-Client des Betriebssystems geben sie das folgende Kommando in der Eingabeaufforderung bzw. einem Terminal ein.

```
1 ssh iotraffic5@192.168.107.1 -N -L8080:localhost:8080
```

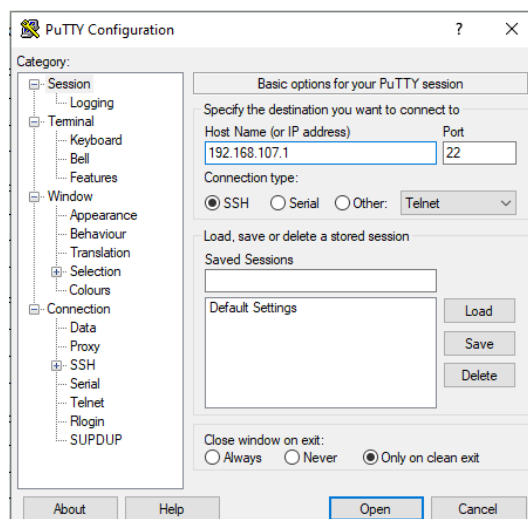
3.4.2 Alternative: Grafischer SSH-Client Putty (Win, Linux)

Um einen SSH-Tunnel über die grafische Oberfläche zu starten öffnen sie zunächst das Programm Putty (siehe folgende Abbildungen). Öffnen Sie im Konfigurationsbaum Connection / SSH / Tunnels. Geben sie bei Source Port 8080 und unter Destination localhost:8080 ein und drücken sie dann auf Add.

Öffnen Sie nun im Konfigurationsbaum Session. Geben sie bei Host Name 192.168.107.1 und drücken abschließend auf Open.



Konfiguration der Tunnelparameter.



Konfiguration des Hosts.

3.4.3 Verbindung bei bestehendem Tunnel

Nach dem erfolgreichen Login, mittels Passwort, wird keine Shell angezeigt aber der Tunnel ist aufgebaut. Somit ist der folgenden Endpunkt auf dem System des Endbenutzers (localhost) verfügbar (siehe Kapitel Konfiguration der Linien im OpenDataCam):

- Konfigurationsoberfläche auf Port 8080 (<http://localhost:8080>)

3.5 Abfrage des Kamera Status

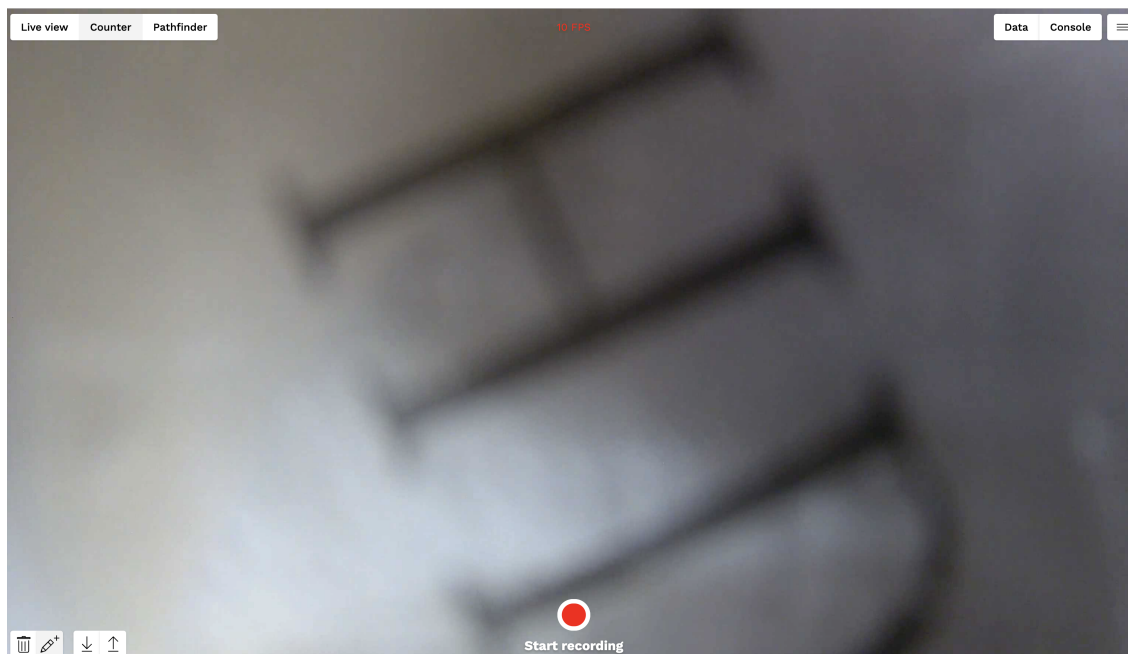
Eine Abfrage des Kamera Status kann auf einem verbundenen System durch aufrufen der URL <http://localhost:8080/status> erfolgen. Im Folgenden eine Beispielausgabe:

```
1  {
2    "counterSummary": {},
3    "trackerSummary": {
4      "totalItemsTracked": 0
5    },
6    "videoResolution": {
7      "w": 3072,
8      "h": 1728
9    },
10   "appState": {
11     "yoloStatus": {
12       "isStarting": false,
13       "isStarted": true
14     },
15     "isListeningToYOLO": true,
16     "recordingStatus": {
17       "requestedFileRecording": false,
18       "isRecording": true,
19       "currentFPS": 10,
20       "recordingId": "6556542f9bde4e071fb9d982",
21       "dateStarted": "2023-11-16T17:41:03.855Z",
22       "filename": ""
23     }
24   }
25 }
```

3.6 Konfiguration der Linien im OpenDataCam

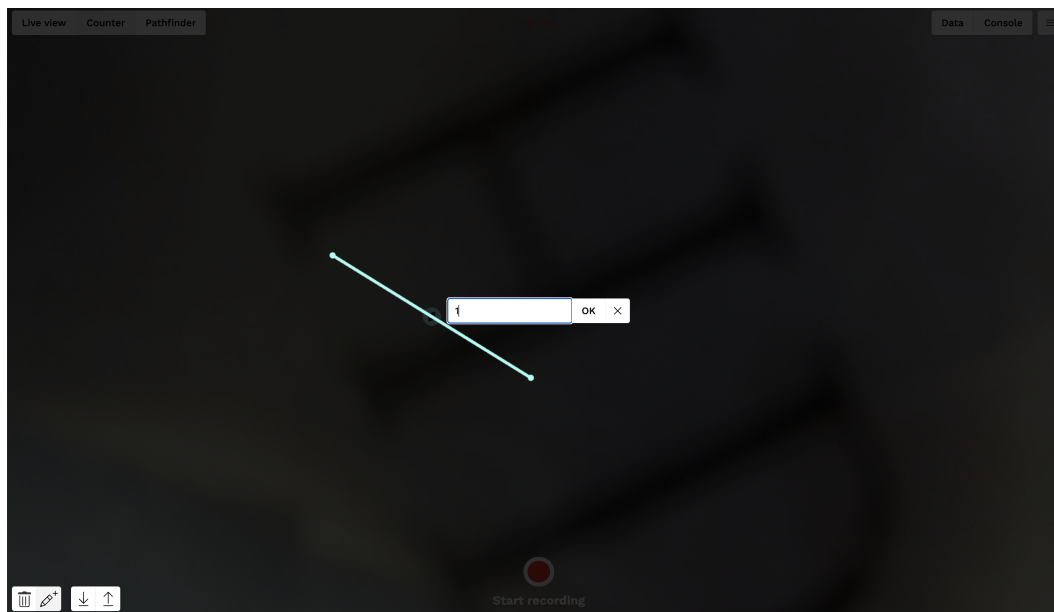
- Aufrufen der folgenden URL im Browser: <http://localhost:8080/status>

Die folgende Ansicht erscheint:



Konfigurationsoberfläche

1. Drücken des Buttons Stop Recording (unten Mitte) um die aktuelle Aufnahme zu unterbrechen.
2. Anklicken von Counter (oben links) um in den Zähler Editor zu kommen.
3. Anklicken des Buttons "Stift, Pluszeichen" zum Hinzufügen einer neuen Linie.
4. Durch Klicken und Ziehen im Bild können nun die Linien hinzugefügt werden.
5. Nach Zeichnen einer Linie wird ein Name vergeben, dieser muss zwingend eine ganze Zahl sein.
Z.B.: 1,2,3,4,5 oder 6.
6. Durch Klicken auf die Pfeile in der Mitte der Linie kann die Zählrichtung (uni- oder bidirektional) eingestellt werden.
7. An dieser Stelle ist es sinnvoll einen Screenshot der Linien zur späteren Verarbeitung zu haben.
Im folgenden eine Beispielsansicht mit gezogenen Linien:



Zeichnen einer

Zähl-Linie in der iotraffic Oberfläche

4 LoRaWAN Payload Beschreibung

Dieser Abschnitt beschreibt den Aufbau der Payload des Produktes **iotraffic** der Firma iotec.

4.1 Uplink

Die **iotraffic** Kamera sendet 3 unterschiedliche Status- und Zähler-Payloads welche im Folgenden beschrieben sind.

4.1.1 Zählraten

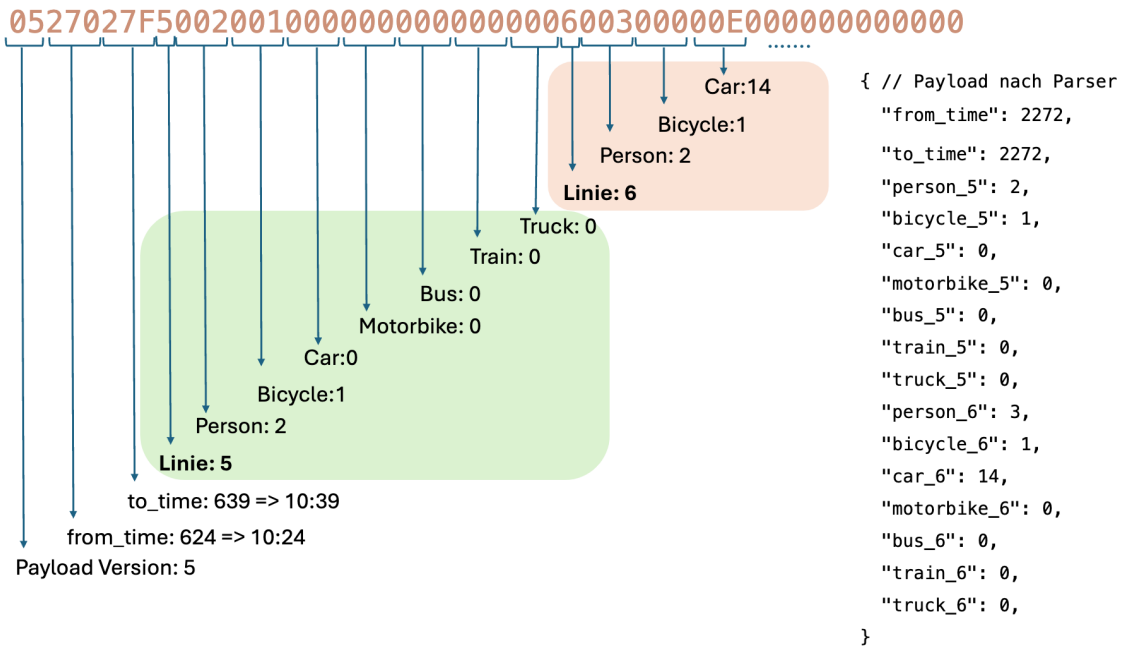
Die Übertragung der Zählraten erfolgt in einem festen Intervall. Das Messintervall kann per LoRaWAN Downlink eingestellt werden. Pro Datenpaket werden die Zählraten von bis zu 4 gemessenen Überquerungslinien übertragen. Wenn mehr als 4 Linien übertragen werden, folgt ein weiteres Paket auf demselben Port. Pro Messlinie und Intervall können maximal 4095 Überquerungen pro Fahrzeugklasse gemessen werden. Wird dieser Wert erreicht sollte das Messintervall (Standard = 15 Minuten), wie im Downlink Abschnitt beschrieben, verringert werden. Da 1-4 Linien übertragen werden sind evtl. fehlende Bytes mit optional (opt.) gekennzeichnet.

- Port: 16
- Encoding: base16 (hex)

Tabelle 1: Zähldatenpaket

Bit	Bits	Datentyp	Variablenname	Beschreibung	Opt.
1	8	UINT18	payload_version	Payload ID zu Auf- und Abwärtskompatibilität bei Neuerungen	
9	12	UINT12	from_time	Startzeit der Messung als Minute des Tages. Startet um 0:00 mit 0, 12:00 ist 720 und 23:55 ist 1435	
21	12	UINT12	to_time	Endzeit der Messung als Minute des Tages	
33	4	UINT4	line_number_1	Die Nummer der Linie der folgenden Zähldaten (z.B. 1 oder 5)	
37	12	UINT12	person_line_1	Anzahl der Fußgänger, die im Messintervall die Messlinie überquert haben	
49	12	UINT12	bicycle_line_1	Anzahl der Fahrräder, die im Messintervall die Messlinie überquert haben	
61	12	UINT12	car_line_1	Anzahl der Autos, die im Messintervall die Messlinie überquert haben	
73	12	UINT12	motorbike_line_1	Anzahl der Motorräder, die im Messintervall die Messlinie überquert haben	
85	12	UINT12	bus_line_1	Anzahl der Busse, die im Messintervall die Messlinie überquert haben	
97	12	UINT12	train_line_1	Anzahl der Züge, die im Messintervall die Messlinie überquert haben	
109	12	UINT12	truck_line_1	Anzahl der Lastkraftwagen, die im Messintervall die Messlinie überquert haben	
121	4	UINT4	line_number_2	s.o.	opt.
125	12	UINT12	person_line_2	s.o.	opt.
137	12	UINT12	bicycle_line_2	s.o.	opt.
149	12	UINT12	car_line_2	s.o.	opt.
161	12	UINT12	motorbike_line_2	s.o.	opt.
173	12	UINT12	bus_line_2	s.o.	opt.
185	12	UINT12	train_line_2	s.o.	opt.
197	12	UINT12	truck_line_2	s.o.	opt.
209	4	UINT4	line_number_3	s.o.	opt.
213	12	UINT12	person_line_3	s.o.	opt.
225	12	UINT12	bicycle_line_3	s.o.	opt.
237	12	UINT12	car_line_3	s.o.	opt.
249	12	UINT12	motorbike_line_3	s.o.	opt.
261	12	UINT12	bus_line_3	s.o.	opt.
273	12	UINT12	train_line_3	s.o.	opt.
285	12	UINT12	truck_line_3	s.o.	opt.
297	4	UINT4	line_number_4	s.o.	opt.
301	12	UINT12	person_line_4	s.o.	opt.
313	12	UINT12	bicycle_line_4	s.o.	opt.
325	12	UINT12	car_line_4	s.o.	opt.
337	12	UINT12	motorbike_line_4	s.o.	opt.
349	12	UINT12	bus_line_4	s.o.	opt.
361	12	UINT12	train_line_4	s.o.	opt.
373	12	UINT12	truck_line_4	s.o.	opt.

4.1.1.1 Beispiel Abbildung 1: Visualisierung des Parsens der Zähldaten-Payload:



4.1.2 Statusdaten AI-Modul

Die Statusdaten des AI Moduls werden zu Evaluationszwecken und zu Remote-Maintenance Zwecken gesendet.

- Port: 17
- Encoding: base16 (hex)

Tabelle 2: Statusdaten AI-Modul

Index	Bytes	Datentyp	Variablenname	Beschreibung
1	1	uint8	payload_version	ID der Payload für Aufwärts- und Abwärtskompatibilität
2	1	uint8	cpu_pct	Aktuelle CPU-Auslastung in Prozent
3	1	uint8	vmem_pct	Aktuelle Auslastung des virtuellen Speichers in Prozent
4	1	uint8	disc_pct	Aktuelle Speicherauslastung in Prozent
5	1	uint8	os_load_1_average	Die durchschnittliche Systemauslastung der letzten Minute
6	1	uint8	os_load_5_average	Die durchschnittliche Systemauslastung der letzten 5 Minuten
7	1	uint8	os_load_15_average	Die durchschnittliche Systemauslastung der letzten 15 Minuten
8	1	uint8	network_if_status	Netzwerk Interface Status (Bitmaske): 1. Bit WiFi, 2. Bit Ethernet (interne Kameraverbindung)
9	1	uint8	ram_pct	Aktuelle Auslastung des Arbeitsspeichers in Prozent
10	1	uint8	swap_pct	Aktuelle Nutzung des Swap-Speichers in Prozent
11	1	uint8	gpu_usage	Aktuelle GPU Auslastung in Prozent
12	1	uint8	cpu_temp	Aktuelle CPU Temperatur in °C
13	1	uint8	gpu_temp	Aktuelle GPU Temperatur in °C
14	1	uint8	error_code	Aktueller Fehlercode
15	4	uint32	voltage	Spannung am AI-Modul in mV
19	4	uint32	timestamp	Zeitstempel des AI-Moduls
23	4	uint32	uptime	Minuten seit dem letzten Neustart des AI-Moduls
27	4	float32 LE	dht_temp	Temperatur im Gehäuse in °C
31	4	float32 LE	dht_humid	Relative Luftfeuchte im Gehäuse in Prozent

4.1.3 Statusmeldungen Steuerungsmodul

Hier werden Statusmeldungen der Steuerungselektronik vom iotraffic Device gesendet. In zwei Bytes wird übertragen was (`stm_what`) aus welchem Grund (`stm_why`) passiert ist.

- Port: 18
- Encoding: base16 (hex)

Tabelle 3: Aufbau Uplink-Paket

Byte	Byte-Laenge	Beschreibung	Wertebereich
1	1	stm_what	Was ist passiert?
2	1	stm_why	Warum/Wodurch ausgelöst?

Tabelle 4: Mögliche Werte für stm_what

Byte	Name	Beschreibung
0x00	loraReset	LoRaWAN Modul hat sich neu gestartet.
0x01	aiReset	AI Modul wurde per Reset-Pin neu gestartet.
0x02	aiShutdown	AI Modul wurde per Befehl über die serielle Schnittstelle heruntergefahren.
0x03	setInterval	Sendeintervall der Messdaten im AI Modul wurde per Befehl über die serielle Schnittstelle auf den gegebenen Wert eingestellt.
0x04	aiReboot	Das AI Modul wurde per Befehl über die serielle Schnittstelle neu gestartet.
0x05	toggleWifiOn	WLAN des AI Moduls wurde per Befehl über die serielle Schnittstelle eingeschaltet.
0x06	toggleWifiOff	WLAN des AI Moduls wurde per Befehl über die serielle Schnittstelle ausgeschaltet.
0x07	sendStats	Das AI Modul wurde per Befehl über die serielle Schnittstelle dazu aufgefordert, die Statusdaten zu senden.
0x08	setTime	Der aktuelle Zeitstempel wurde per serieller Schnittstelle an das AI Modul geschickt.
0x09	aiStart	Das AI Modul wurde per Start-Pin eingeschaltet.

Tabelle 5: Mögliche Werte für stm_why

Byte	Name	Beschreibung
0x00	downlink	Ein Befehl ist per Downlink aus dem LoRaWAN-Portal gekommen.
0x01	comTimeOut	Kommunikations-time-out: Die letzte erfolgreiche Datenübertragung vom AI- zum Steuerungsmodul über die serielle Schnittstelle überschreitet die erlaubte Zeit.
0x02	ackTimeOut	ACK-time-out: Die letzte Datenübertragung vom Steuerungs- zum AI Modul über die serielle Schnittstelle wurde nicht innerhalb der vorgegebenen Zeit quittiert.
0x03	jetsonRebootAck	Das AI Modul hat das Reboot-ACK nach seinem Neustart geschickt.
0x04	resetCauseUnknown	Das Steuerungsmodul wurde aus unbekanntem Grund neu gestartet.
0x05	resetCauseLowPower	Das Steuerungsmodul wurde neu gestartet, weil die Versorgungsspannung zu niedrig war.
0x06	resetCauseWindowWatchdog	Das Steuerungsmodul wurde neu gestartet, weil der Watchdog ausgelöst hat.
0x07	resetCauseIndipendentWatchdog	Das Steuerungsmodul wurde neu gestartet, weil der Watchdog ausgelöst hat.
0x08	resetCauseSoftware	Das Steuerungsmodul wurde neu gestartet, weil der Befehl dazu im Programmcode ausgelöst wurde.
0x09	resetCausePowerCycle	Das Steuerungsmodul wurde neu gestartet, weil die Versorgungsspannung unterbrochen wurde.
0x0A	resetCauseResetPin	Das Steuerungsmodul wurde neu gestartet, weil der Reset-Pin aktiviert wurde.

4.2 Downlinks

4.2.1 Paketstruktur Payload senden

Jedes Datenpaket hat die folgende Struktur:

- Start-Marker (1 Byte): Ein festes Byte, um den Anfang eines neuen Pakets zu kennzeichnen: 0xAA .
- Befehl oder Datentyp (1 Byte): Ein Byte, um den Befehl oder den Datentyp des Pakets zu identifizieren.
- Nutzdatenlänge (1 Byte): Ein Byte, die die Länge der Nutzdaten in Bytes angeben (maximal wert 51 für 51 Bytes Nutzdatenlänge).
- Nutzdaten (variable Länge, max 51): Die eigentlichen Daten des Pakets, deren Länge in dem vorherigen Byte angegeben wurde.
- Prüfsumme (1 Byte): Ein Byte, das verwendet wird, um die Integrität des Pakets zu überprüfen. (Checksumme / Modulo-N-Summe)

4.3 Befehle und Datentypen

Die Befehle oder Datentypen können durch verschiedene Byte-Werte repräsentiert werden:

4.3.1 Downlinks zum AI-Modul

- 0x01: Befehl "reset" - Neustart des Systems durch Reset Pin (eventuelle Software Vorkehrungen können zuvor gestartet werden)
- 0x02: Befehl "shutdown" - Shutdown des Systems auf Software-Ebene, z.B. zur Demontage
- 0x03: Befehl "setInterval" - Interval setzen (mit 1 Byte Parameter, Minuten)
- 0x04: Befehl "reboot" - Software Neustart des Jetson
- 0x05: Befehl "toggleWifiOn" - WiFi aktivieren für + 1-byte mit Wert 0x1E zur Auswahl des WiFi
- 0x06: Befehl "toggleWifiOff" - WiFi deaktivieren
- 0x07: Befehl "sendStats" - Extrapaket mit Status senden.
- 0x08: Befehl "setTime" - LoRaWAN Timestamp als payload (uint32 unix seconds)
- 0x09: Befehl "start" - Start des Systems durch Power Pin

4.4 Beispiel für ein Datenpaket mit Befehl "setInterval"

- Start-Marker: 0xAA
- Befehl: 0x03 (setInterval)
- Nutzdatenlänge: 0x01 (1 Byte Nutzdaten)
- Nutzdaten (Parameter): 0x05 (z.B. 5 Minuten Intervall)
- Prüfsumme: $(0xAA + 0x03 + 0x01 + 0x05) \% 256 = 0x03$
- Das gesamte Paket wäre also: 0xAA, 0x03, 0x01, 0x05, 0x03

Tabelle 6: Beispiel Downlinks für eine Integration in das Portal

Kommando	Beispielpayload	Beschreibung
reset	AA0100AB	Reset des Systems
shutdown	AA0200AC	Shutdown des AI-Moduls
setInterval_10min	AA03010AB8	Messintervall auf 10 Minuten setzen
setInterval_15min	AA03010FBD	Messintervall auf 15 Minuten setzen
reboot	AA0400AE	Reboot des AI-Moduls
toggleWifiOn	AA05011ECE	WiFi temporär aktivieren
toggleWifiOff	AA0600B0	WiFi deaktivieren
sendStats	AA0700B1	Aktuelles Statuspaket senden
setTime	AA0800B2	Uhrzeit neu durch die LoRaWAN Zeit setzen
start	AA0900B3	Start des AI-Moduls