

Albatros

Continental Tire Pressure

Inhoudsopgave

[Inhoudsopgave 2](#_Toc100732433)

[1. Documentdata 3](#_Toc100732434)

[1.1 Versions 3](#_Toc100732435)

[1.2 References 3](#_Toc100732436)

[1.3 Distributionlist 3](#_Toc100732437)

[3. Background 4](#_Toc100732438)

# Document data

## Versions

|  |  |  |
| --- | --- | --- |
| Datum | Versie | Opmerkingen |
| 13-4-2022 | 0.1 | Eerste Versie |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## References

|  |  |  |
| --- | --- | --- |
| Datum | Versie | Opmerkingen |
| api\_public\_documentation\_final.pdf | 1.0 | ContiConnect Live via Api  Description on how to upload tire pressure data to the conticonnect website. |
| CPC-3rd-Party-Connection\_v0160.pdf | 1.6 | Description of the content of the sensor data.  (Jörg Hanna) |
|  |  |  |
|  |  |  |
|  |  |  |

## Distributionlist

|  |  |  |
| --- | --- | --- |
| Datum | Versie | Opmerkingen |
|  | 0.1 | Eerste Versie |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Abbreviations

|  |  |
| --- | --- |
| Datum |  |
| CAN | Controller area network. (inside the vehicle) |
| CCU | Central Control Unit |
| CPC | Continental Pressure Check ™ |
| TTM | Truck Tire Module |
| PGN | Parameter Group Number |

# Background

A number of vehicle has been equipped with tire pressure sensors by Continental.

These sensors are connected to a central device within the vehicle and this device is connected to the Pilotfish Vehiclegateway. (via the vehicle’s CAN bus)

The sensors communicate information like pressure and temperature, but also ‘alarms’, for instance when pressure is lost quickly or pressure is below a threshold.

The Vehiclegateway stores the collected data in a Azure Blob Storage.

Continental provides a website where the condition of all vehicles and tires is monitored and managed.

This document is the description of the software to collect the data from the Azure Blob Storage and deliver it to the Continental website.

## Collecting the data

Every second this data is collected by the pilotfish, and written to a Azure Blob Storage about every 10 minutes in the form of a blobitem containing up to about 600 payloaditems.

Each payload item contains many messages containing information (PGN’s) about one or more tires of the vehicle. Not every tire is represented in every payload.

Every payload contains all CAN data collected, during one second. Beside the tiresensors information, more CAN data is found in the payload such as engine’s rpm, fuel level, acceleration, breaking etc.

## Processing step 1, divide and conquer

Every blobitem is stored on the exact same location in the Azure Blob. This causes a performance issue when processing the data. Therefore the first step in processing the blobitems is moving them to a subdirectory structure containing year/month/day/vehiclenumber. This step is performed by the BlobDistributerService.

## Processing step 2, update continental website

Completely independent of step 1, the content of the blobitems is processed.

The data is read and send to the continental backoffice website using an API.

For every blobitem, every payload is read, data is send to continental:

1. At least once every 10 minutes
2. When any alarm value changes compared to the latest send data
3. When the tire pressure changes by 10% or more compared to the latest send data
4. When the temperature changes by 10% or more compared to the latest send data

# Technical details

## SAE J1939

The low level data structure confirms to the SAE J1939, the open standard for the Society of Automotive Engineers for Controller Area Network (CAN) bus data.

SAE J1930 describes messages put on the CAN, where every message is identified by a Parameter Group Number (PGN).

PGN’s are identified by a 4 digit hexadecimal code or a decimal number. The important PGN’s used by Continental are described in the document “CPC-3rd-Party-Connection\_v0160.pdf” called PGN doc below.

The vehicles Central Control Unit (CCU) can be configured to put a selection of the PGN’s on the CAN [ TBD : is this right? How to we to this? ]

We need to combine the information in different PGN’s to find out the vehiclenumber, SensorId, and the location of the sensor within the vehicle.

When a blobitem is analysed, and thus all payloads are analysed, we find that not all

payload’s contain all PGN’s and if a payload contains certain PGN’s they may or may not be present for all tires.

These are the used PGN’s

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PGN | Documentation | Description | Identifiers / relational fields | Content |
| 0xFF04 | paragraph 4.9 | CPC Graphical Position  Main source for knowing what sensor has what id and is located where, and for relating the 3 PGN’s below to each other | TireId  TireLocation  TTM-ID = SensorID | Graphical Position |
| 0xFF02 | Paragraph 4.8 | TTM Data. Summary  Alarms a little different from the 0xFEF4 | TireId | TirePressure, temperature and alarm data. |
| 0xFEF4 | Paragraph 4.3 | Tire Condition | TireLocation | Pressure Temperature and alarms. |
| 0xFC42 | Paragraph 4.4 | Tire Condition 2 | TireLocation | Pressure and Required pressure |

# Distribution of Blobs

, making it extremely slow to process. (in September 2021, when about 90 vehicles were connected, we found that querying the Azure Storage increased by 5 to 20 minutes every day

, for instance

arriva-nl/cloud-fms/9601\_cloudfms1-2\_20220412121845\_2232834\_1649766530215.json

thus