ZMQ Message Format

Message Types

- Command Messages
    - Drive Command
    - Camera Command
- Video Messages
- Sensor Messages

Command Messages

A ZMQ Command Message consists of 2 Frames:

- Robot ID
- Data

The **Robot ID** always has to be the first Frame. It is used as a Subscribe-Filter to determine from which Robot the message was sent or for which Robot the message is intended.
The **Data** Frame is the frame in which the actual message data is stored. The data depends on the message type.

The data of a Command Message is sent as a JSON String. Every Command Message has a **header** field and a **data** field. The format of the header is the same for every command message and consists of the following subfields:

- **Header ID.** The header ID defines the type of the command message.
  Type: int
  Valid Values:
    - 171 (Drive Command)
    - 172 (Camera Command)
- **TID.** The Transaction ID (or message nr). It is incremented for every sent message and counts over all command messages.
  Type: int
- **Timestamp.** The current system time in milliseconds since January 1, 1970 00:00:00 UTC
  Type: long
- **Robot ID.** The Robot ID used to identify the robot. It is the same ID as used in the first frame of the ZMQ Message.
  Type: String
- **Version.** The JSON Protocol Version used to create the command. The Version will be

adjusted every time the JSON format is changed and will be used to correctly decode a JSON String.
Type: String

Example:

```
"header" : {
    "timestamp" : 1363353888309,
    "id" : 171,
    "robot" : "Romo",
    "tid" : 0,
    "version" : "0.1"
}
```

Drive Command

A Drive Command is used to drive the robot around. The data field consists of the following subfields:

- **Speed.** Defines the speed in % with which the robot should drive
  Type: double
  Valid Values: 0 - 100 [%], Exception: a value of -1 will let the robot choose it's own default speed.
- **Radius.** The radius in mm of the turn which should be executed
  Type: double
  Valid Values: 1 - 1000 [mm], Exception: a value of 0 will let the robot drive straight
- **Move.** The type of move which should be executed.
  Type: String
  Valid Values:
  - FORWARD
    A forward move.
  - STRAIGHT_FORWARD
    A forward move, parameter radius will be ignored.
  - BACKWARD
    A Backward move
  - STRAIGHT_BACKWARD
    A backward move, parameter radius will be ignored
  - ROTATE_LEFT
    Rotate left / counter clockwise
  - ROTATE_RIGHT
    Rotate right / clockwise
  - NONE
    Stop the current move.

Example:

```
"data" : {
        "speed" : -1,
        "angle" : 0,
        "move" : "STRAIGHT_FORWARD"
}
```

Camera Command

A Camera Command is used to control the robot's / smartphone's camera. It's data field consists of one subfield:

- **Type.** Defines the type of the command
  Type: String
  Valid Values:
  - TOGGLE
    Toggles between the available cameras
  - ON
    Turns the camera on
  - OFF
    Turns the camera off

Example:

```
"data" : {
        "type" : "TOGGLE"
}
```

Video Message

Video is streamed frame by frame a series of JPEG images. The Camera image is compressed to JPEG. A Video Message holds exactly one Video Frame. A ZMQ Video Message consists of 3 Frames:

- Robot ID
- Header
- Data

The **Robot ID** always has to be the first Frame. It is used as a Subscribe-Filter to determine from which Robot the message was sent or for which Robot the message is intended.
The **Header** Frame stores a JSON String and consists of the following fields:

- **Header ID.** The header ID defines the type of the command message.
  Type: int
  Valid Values:

        - 181 (Raw Video)
- **TID.** The Transaction ID (or message nr). It is incremented for every sent message and counts over all video messages.
  Type: int
- **Timestamp.** The current system time in milliseconds since January 1, 1970 00:00:00 UTC
  Type: long
- **Robot ID.** The Robot ID used to identify the robot. It is the same ID as used in the first frame of the ZMQ Message.
  Type: String
- **Rotation.** Camera Images on an android smartphone are oriented in landscape mode. We assume however that the smartphone is used in portrait mode. So in order to display the Video Frame correctly on the screen, the Video has to be rotated by the given value.
  Type: Integer
  Valid Values: -360 - 360 [°]
- **Version.** The JSON Protocol Version used to create the command. The Version will be adjusted every time the JSON format is changed and will be used to correctly decode a JSON String.
  Type: String

Example:

```
"header" : {
        "timestamp" : 1363353888309,
        "id" : 171,
        "robot" : "Romo",
        "tid" : 0,
        "rotation" : 0,
        "version" : "0.1"
}
```

The **Data** Frame is the frame in which the actual video frame is stored as an rgb byte array. Every Command Message has a **header** field and a **data** field. The format of the header is the same for every command message and consists of the following subfields:

Sensor Message

The data frame of a Sensor Message is sent as a JSON String which consists of a header and a data field, similar to the Command Messages. The header field looks the same as the header of the Command Messages. As Header ID, the following Values are valid:

- 191 (Sensor Message)

The Data field is of the type JSON list, where each element has the following structure:

- **Sensor Name**. The name of the sensor. Has to be unique.
  Type: String
- **Sensor Type.** The datatype of the sensor value
  Type: String
  Valid Values: "String", "Integer", "Double"
- **Sensor Value.** The value of the given sensor
  Type: defined by the field **Sensor Type**

Command and Sensor Message:

ROBOT ID	**DATA**
ROBOT ID	**Header**: HEADER ID; TRANSACTION ID; TIMESTAMP; ROBOT ID; VERSION **Data**: Drive Command: MOVE; SPEED; RADIUS Camera Command: TYPE Sensor Message: NAME, NAME, NAME, ... / TYPE, TYPE, TYPE, ... / VALUE, VALUE, VALUE, ...

Video Message:

ROBOT ID	**HEADER**	**DATA**
ROBOT ID	HEADER ID; TRANSACTION ID; TIMESTAMP; ROBOT ID; ROTATION; VERSION	JPEG RGB Array