Kommunikation

Um genügend Rechenleitung für das Auswerten der Kamera und das Steuern der Sensoren und Aktoren zu haben, werden zwei Raspberry Pis eingesetzt. Somit werden auch zwei eigenständige Programme laufen. Ein Programm wird für die Auswertung der Kamera und der Steuerung des gesamten Ablaufs zuständig sein. Das zweite Programm wird für die Steuerung der Sensoren und Aktoren zuständig sein.

Die beiden Programme werden über eine Netzwerkverbindung untereinander Daten austauschen. Damit dies möglich ist, wurde ein Kommunikationsprotokoll definiert.

# Protokoll

|  |  |
| --- | --- |
| Message ID | Message Type |
| Timestamp |  |
| Payload | |

**Message ID**: Inkrementierender Wert (Integer)

**Message Type**: ID für die Art der Nachricht (#REF Kapitel) (Integer)

**Time Stamp**: Wann die Nachricht abgeschickt wurde (Unix Time Stamp)

**Payload**: Je nach Art der Nachricht verschieden

## Message Types

Der Message Type ist aus einer Zahl mit 4 Ziffern aufgebaut.

|  |  |  |  |
| --- | --- | --- | --- |
| Board Identifier | Command Group Identifier | Command ID | |
| 0-9 | 0-9 | 0-9 | 0-9 |

**Board Identifier**: Identifiziert das Board (Raspberry Pi) welches die Nachricht gesendet hat.

**Command Group Identifier**: Jede Nachricht gehört zu einer Gruppe. (Drive, Garbage  
Collection, Garbage Disposal, System State Messages)

**Command ID**: Identifiziert das Kommando.

### Messages

# Insert messages here

## Payloads

# Insert Payloads here