# Schnittstelle Arduino-Fernwartung

Der Datenaustausch zwischen Controllino-MAXI und NodeMCU wird mithilfe der seriellen Schnittstelle UART realisiert. Die hierfür in der Arduino-Bibliothek zur Verfügung stehenden Funktionen erleichtern die Realisierung einer solchen Übertragung.

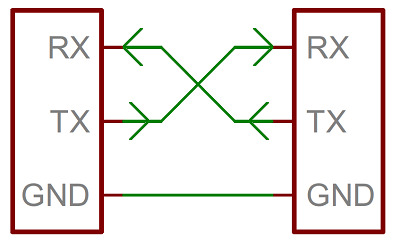


Abbildung : Verschaltung zwischen Controllino-MAXI und NodeMCU

Um eine serielle Schnittstelle zwischen zwei Mikrocontrollern zu realisieren, ist es lediglich notwendig die RX und TX-Pins beider Controller gekreuzt sowie beide GND-Pins miteinander zu verbinden.

## Das Telegramm

Eine Datenübertragung erfordert ein gemeinsames Protokoll zwischen Sender und Empfänger. Da in diesem Fall Sender und Empfänger von zwei unterschiedlichen Teams bearbeitet werden ist eine gute Schnittstellen Koordination erforderlich.

Beide Seiten der Übertragung müssen auf der gleichen Baudrate kommunizieren, damit die übertragene Nachricht korrekt eingelesen werden kann.

Darüber hinaus ist es notwendig ein festen Telegramminhalt zu vereinbaren, um beim Einlesen der Nachricht eine Korrekte Interpretation der Daten durchführen zu können.

Aufgrund der statischen Interpretation der Werte ist eine konstante Länge der übertragenen Nachricht notwendig.

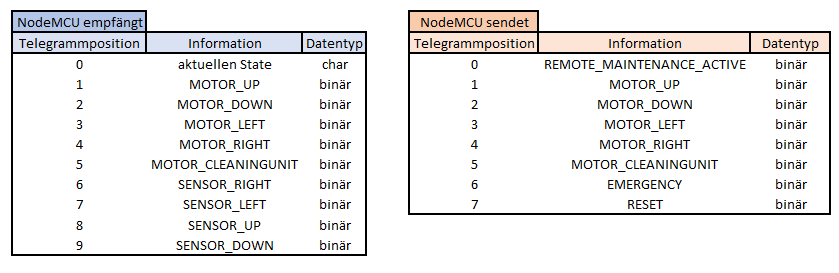


Abbildung : Übertragungsprotokoll aus Sicht des NodeMCU

Das Übertragen des aktuellen Systemzustandes stellt ein Problem dar. Systemintern wird der Zustand einem Integer Wert von 0-11 zugewiesen. Da dieser Zustand zwischen 1 und 2 Zeichen variiert, werden zu Zustande als ASCII-Zeichen anhand der gezeigten Tabelle interpretiert.

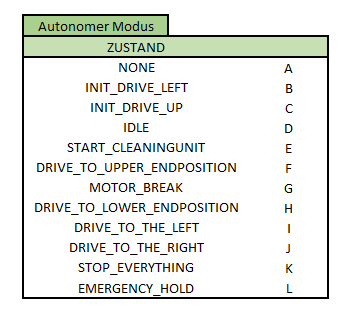


Abbildung : Codierung der empfangenen Zustände

Ein Ausblick auf eine mögliche Verbesserung der Übertragung wäre die Verwendung von Trennzeichen. Dadurch könnten zusammenhängende Nachrichten mit variierenden Längen besser interpretiert werden und eine fehlerhafte Übertragung wird leichter erkannt.

## Senden der Nutzereingaben

Die vom Nutzer an der GUI getätigten Eingaben werden Systemintern in einem Array aus acht Elementen gespeichert.

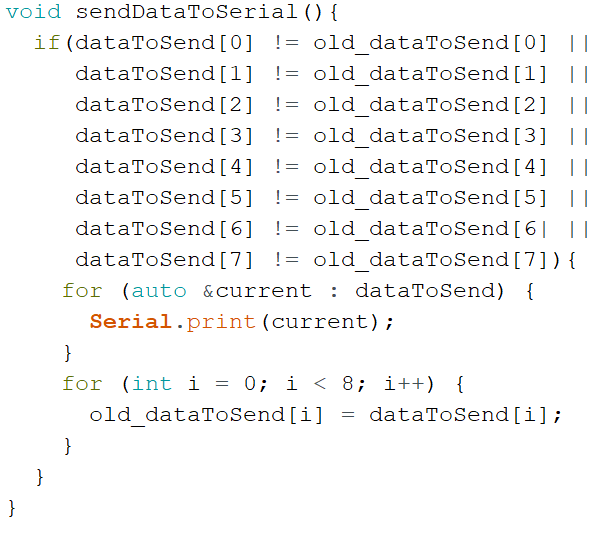


Abbildung : Funktion zur Datenübertragung an den Controllino-MAXI

Dieses Array wird nun überprüft, ob es sich im Vergleich zum letzten Systemtakt verändert hat. Ist dem der Fall, so wird durch das Array iteriert und mithilfe der Funktion *Serial.print()* werden die einzelnen Elemente des Arrays über den Seriellen Ausgang übertragen.

## Empfangen der Übertragung aus Controllino-MAXI

Eine Übertragung von Controllino-MAXI findet ebenfalls nur statt, wenn sich der Zustand eines der Telegrammelemente verändert. Beim Empfangen der Nachricht ist es also sinnvoll zunächst mit der Funktion *Serial.available()* zu überprüfen, ob auszulesende Daten am seriellen Eingang anliegen.



Abbildung : Funktion zum Empfangen der Daten von Controllino-MAXI

Liegen Daten am seriellen Eingang an, so werden diese zur Aufbereitung in einem Buffer gespeichert. Das erste eingelesene Element ist laut Telegrammvereinbarung immer ein Buchstabe. Dieser wird als ASCII-Zeichen übertragen. Bei der Interpretation der Empfangenen Nachricht müssen die ersten beiden Stellen des Arrays zunächst in ein Integer Wert gewandelt werden, um diesen als Char darstellen zu können.

Im Anschluss daran, können die eingelesenen Werte einem Array zugewiesen werden, welches zur weiteren Darstellung in der GUI verwendet wird.

## Testen der Funktionen

Das Testen der Funktion von GUI und Übertragung kann mithilfe eine Mock-Ups des AntidustA geschehen. Hierfür wird ein zusätzlicher Arduino verwendet, der nach einem vordefinierten Ablauf das Verhalten des AntidustA während eines Reinigungsvorgangs simuliert.

Dafür wird die Übertragene Nachricht unter Bezugnahme des Hardwareverhaltens in festgelegten Abständen variiert. In der GUI kann so der simulierte Reinigungsvorgang beobachtet werden.

Wechselt man in den manuellen Betrieb des AntidustA, so regiert der Mock-Up auf Nutzereingaben nach vorher definierten Verhaltensweisen.

Auf diese Weise kann die korrekte Funktion der GUI und das Übertragen von Daten an und von dem Controllino-MAXI getestet werden, ohne dass eine Hardware vorhanden ist.