**Kommunikation zwischen ESP und Stepper Arduino**

* Der ESP kommuniziert nur die Nr. der Position, die angefahren werden soll
* Arduino Nano hat den internen Stationszähler immer gespeichert im EPROM und errechnet die Positionsdifferenz, die er zurücklegen
* Anfangs wird kurz die STations-Nummer übermittelt und die aktuelle Position vom Arduino an den ESP geschrieben (damit im Falle eines Ausfalles des ESP’s die Stationen wieder synchron sind)
* Bei einem Trigger durch den Readkontakt wird sowohl Arduino als auch ESP getriggert, um die Kommunikation zwischen den beiden Modulen möglichst auf einem niedrigen Level zu halten
* Dennoch muss die Library die interne Clock des Nanos steuern!

**Anfahrt von Stationen**

* Es wird anfangs berechnet, auf welche Geschwindigkeit beschleunigt werden soll, abh. von der Anzahl Stationen und den dort geltenden Geschwindigkeiten
* Es wird berechnet, bei welchen Stations-Nr. welche Geschwindigkeiten gelten
* Eine Funktion von Jan macht die Beschleunigungskurve von der aktuellen Geschwindigkeit auf die neue

**Interrupts**

* Am Anfang und Ende der Bahn werden auf der Hochbahn Taster betätigt, die Interrupts auf dem Arduino zur Motorsteuerung auslösen
* Dafür werden vom Ultraschallsensor-Arduino nur zwei Interrupts angesteuert.

**Ändern in der Doku:**

* Interrupts am Ultraschallsensor
* Taster als zusätzlicher Interrupt an Motorsteuerung