## Der Motortreiber

Zur Nachführung des Solarmoduls „TPS-103 4W“ zur Sonne ist die Solareinheit drehbar montiert. Mit Hilfe eines Getriebemotors wird diese Plattform je nach Uhrzeit und GPS-Position zur Sonne ausgerichtet. Da die Ausgänge des Arduino Due keine großen Lasten treiben können wird ein Motortreiber-IC benötigt. Die Wahl fiel auf den Motortreiber L293E.

### Einleitung

Der Motortreiber L293E ist in einem DIL 20 Gehäuse aufgebaut. Die Pinbelegung ist in Abbildung 1 dargestellt. Da sich der Sonnenstand im groben um 180° ändert pro Tag ändert muss eine Möglichkeit gefunden werden, den Getriebemotor in zwei Richtungen drehen zu lassen. Dies wird benötigt um das Solarpanel am Abend wieder zur Position „morgens“ zu drehen.

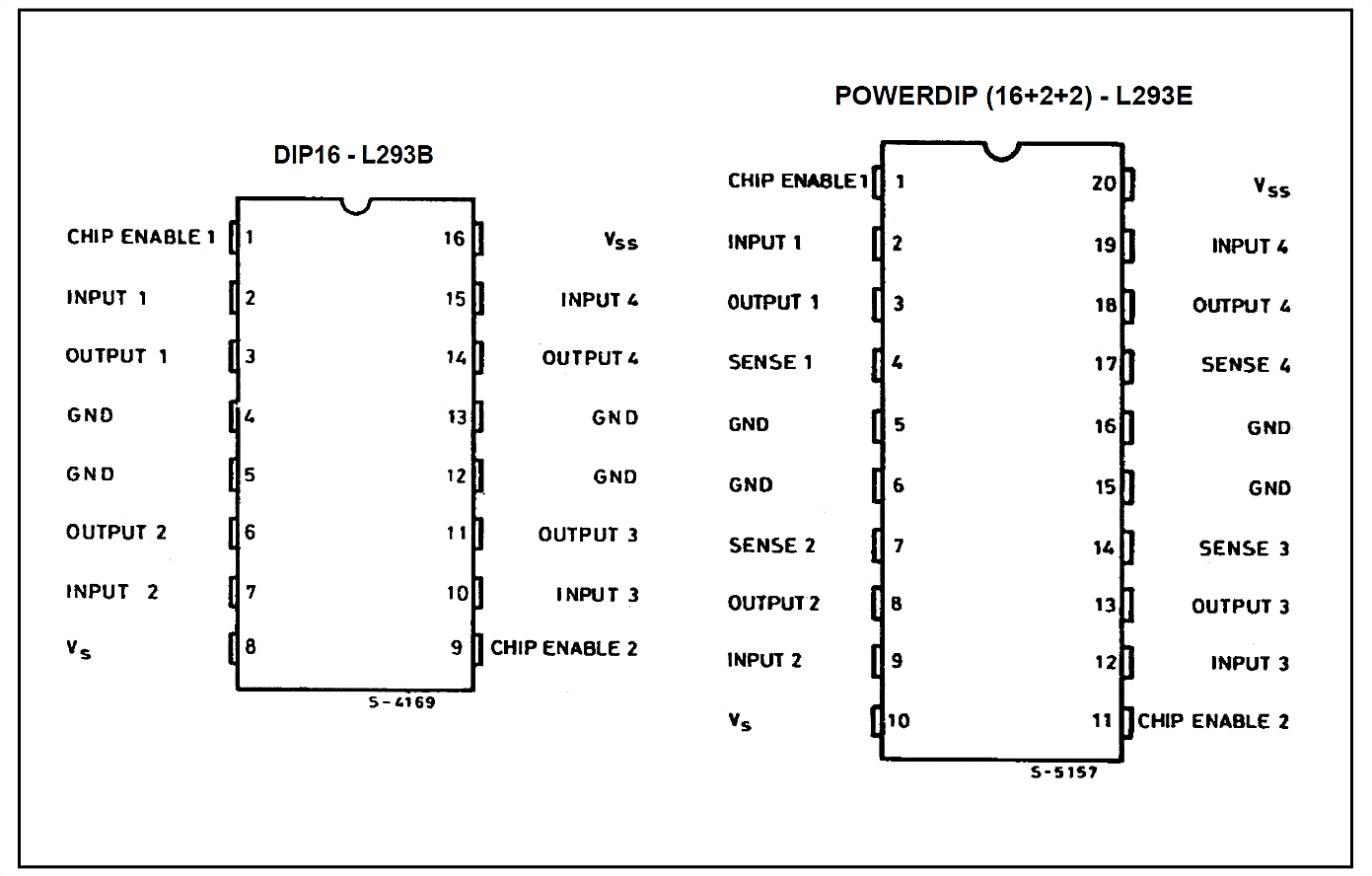


Abbildung Pinbelegung des Motortreiber L293E (Datenblatt L293E)

Der Motortreiber kann in verschiedenen Betriebsmodi genutzt werden. In Abbildung 2 sind diese dargestellt. Aus der Abbildung kann erkannt werden, dass für einen links/rechts Betrieb zwei der vier Ausgänge genutzt werden müssen. Mittels eines Chip-Enable Signals an Pin 11 werden die Ausgänge an Pin 13 und 14 freigeschaltet. Um eine Drehrichtung zu erzeugen müssen die Pins 12 und 19 jeweils komplementär zueinander auf 0 oder 1 gelegt werden, dies erzeugt dann einen Stromfluss für die links/rechts Drehung des Motors.

Bei ersten Tests stellte sich heraus, dass eine ungeregelte Ansteuerung des Getriebemotors zu einer hohen Drehgeschwindigkeit führt. Um eine genaue Positionierung des Solarmoduls zu ermöglichen, muss die Drehgeschwindigkeit des Motors verringert werden. Dies lässt sich durch eine gepulste Ansteuerung der Inputpins 12 und 19 erreichen. Diese Pulsweitenmodulation ist im Abschnitt Pulsweitenmodulation genauer beschrieben.

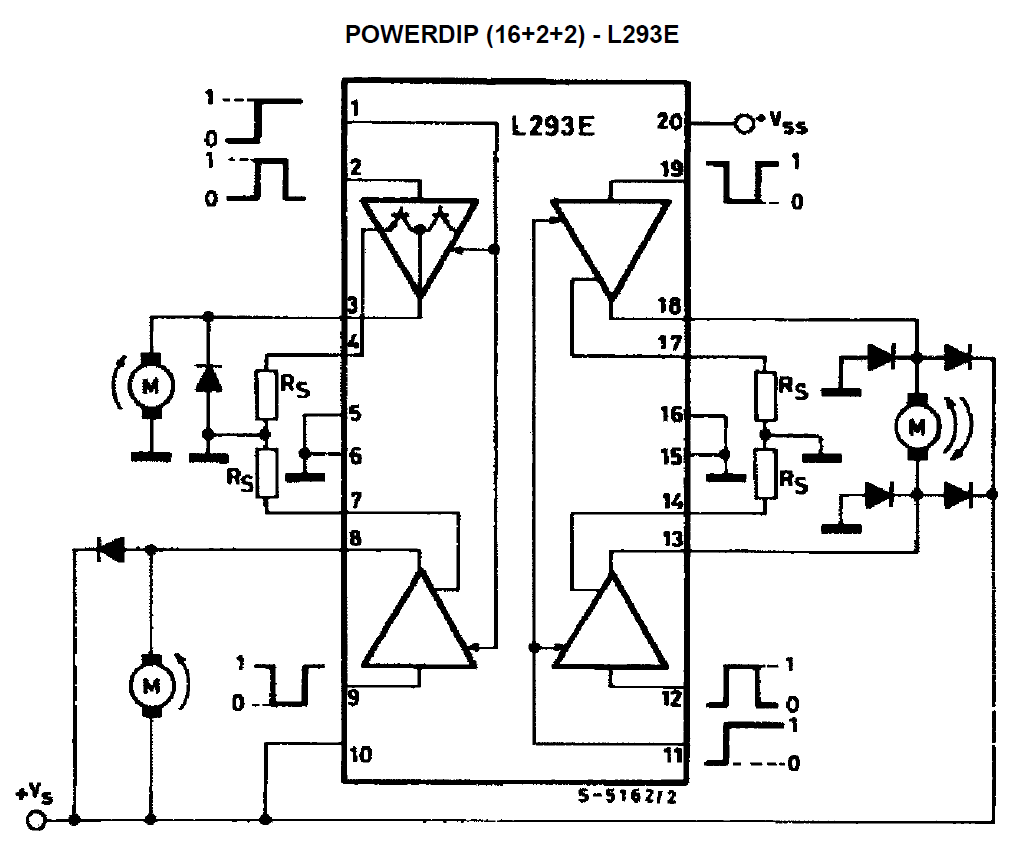


Abbildung Betriebsmodi des Motortreibers (Datenblatt L293E)

### Pulsweitenmodulation

Der Arduino Due erlaubt über die seine Pins 2 bis 13 ein 8-Bit pulsweitenmoduliertes (PWM) Signal auszugeben. In Abbildung 3 ist die Funktionsweise des PWM-Signals dargestellt. Mit Hilfe der Arduino-Funktionen analogWrite() wird festgelegt, in welchem Verhältnis die Einschalt- zur Ausschaltdauer steht. Der Übergabewert der Funktion entspricht einem 8-Bit Wert von 0 – 255, wobei ein höherer Wert für eine längere Einschaltdauer steht.



Abbildung Erläuterung der PWM